Universidad Nacional de La Plata

Facultad de Informática

Framework para la Creación de Aplicaciones de Ciencia Ciudadana, en Google Maps, con Elementos de Gamification

Tesina de Licenciatura en Sistemas

Alumnos: Darío Ezequiel Claramunt y Jorge Agustín Marisi

Director: Alejandro Fernández

Codirector: Diego Torres

Agradecimientos

Queremos agradecer a nuestras familias por haber estado siempre presentes y por habernos motivado y ayudado a buscar un futuro mejor.

También le queremos agradecer a nuestras respectivas parejas, Antonela y Azul, por habernos dado su apoyo y comprensión en todos los ámbitos de la vida desde hace muchos años.

Por último, pero no por eso de menor importancia, queremos agradecerles especialmente a Alejandro y a Diego porque fueron una ayuda y una guía constante durante el desarrollo de esta tesis.

Índice General

1. Introducción

La ciencia, frecuentemente requiere que el científico tome muestras. Un biólogo toma muestras del ecosistema. Un hidrólogo mide la altura de un río en distintos lugares y momentos (toma muestras). Un arquitecto toma fotos, al igual que un astrónomo. Las muestras necesarias pueden ser unas pocas, o miles, dependiendo de la complejidad del fenómeno a estudiar.

La *Ciencia Ciudadana* es una técnica de investigación que permite que personas sin formación específica en el campo científico sean capaces de realizar un aporte. Una de las formas de contribuir de los voluntarios en Ciencia Ciudadana es la toma de muestras, como los casos de *The Great Backyard Bird Count*[1], donde los voluntarios recorren distintas zonas fotografiando aves con el objetivo de armar un catálogo anual con la ubicación de cada especie, *IDAH$_2$O*[2], dedicado a la toma de muestras de la calidad del agua de los flujos de agua en el estado de Idaho (USA), y *Seagrass Spotter*[3], que provee una aplicación para la toma de imágenes de praderas marinas para concientizar y fomentar su conservación.

1.1. Motivación

Este trabajo surge a partir de la consideración de que, para la realización de ciertas tareas de análisis e investigación, tanto en el ámbito académico como a nivel empresarial, es necesario contar con la información de geolocalización de determinados puntos de interés. Ejemplos de tales puntos de interés podrían ser: para un agrónomo la localización de distintas especies de árboles o plantas y para un arquitecto la localización de edificios con valor histórico. Existen también antecedentes de la utilización de este tipo de información respecto de desastres naturales [Horita2013].

En general, la problemática radica en lo extenso de los territorios a explorar y lo exhaustiva de la tarea, que la hace imposible de realizar por una sola persona o un grupo reducido de investigadores, ya que, en muchos casos, requiere una revisión palmo a palmo de los mapas, si es que existen y contienen el nivel de detalle necesario para la tarea a realizar, o un recorrido físico por las regiones a explorar, con el objetivo

[1] http://gbbc.birdcount.org
[2] http://www.uidaho.edu/extension/idah2o
[3] https://seagrassspotter.org

de capturar mediante fotografías u otros medios la información necesaria. Es por esto que se puede recurrir a técnicas de *Crowdsourcing* [Hosseini2015] para dividir el problema de modo tal que pueda ser resuelto por grupos de personas externas a la organización que necesita los datos, que tengan la voluntad de ayudar y que puedan tener conocimientos ya adquiridos de las áreas en cuestión para hacer más rápida la detección.

Nos interesa particularmente el mundo de las muestras digitales (es decir, muestras de datos obtenidas por medio de sensores o cámaras). Existen muchos ejemplos de aplicaciones de Ciencia Ciudadana que, mediante aplicaciones para dispositivos móviles, permiten a voluntarios tomar y enviar muestras (algunas de ellas ya fueron mencionadas). Estas aplicaciones fueron desarrolladas a medida, para un proyecto científico en particular. A pesar de las similitudes que uno pueda observar entre ellas, no existen en la actualidad librerías o frameworks que simplifiquen su desarrollo para nuevos proyectos o escenarios.

Si bien *Zooniverse*[4] ofrece una herramienta para la construcción de aplicaciones de Ciencia Ciudadana, los proyectos creados mediante esta plataforma tienen principalmente como finalidad la clasificación de fotografías o muestras que se encuentran ya a disposición del científico. Por el contrario, al momento de desarrollar este trabajo, no existen herramientas que permitan generar aplicaciones que sean de utilidad para la toma de muestras. Sin este tipo de herramientas, cada científico debe desarrollar su propia aplicación desde cero, con la consideración de que debe abarcar la mayor cantidad de plataformas o sistemas operativos móviles posibles, para alcanzar a la mayor cantidad de voluntarios. Inevitablemente, esto incrementa los tiempos de desarrollo, debido a las particularidades de cada plataforma específica. Con la creación de un framework o librería que reduzca estos tiempos, proveyendo un punto de inicio en común, los científicos pueden enfocarse más rápidamente en la obtención de voluntarios para la toma de muestras, lo que se traduce en un ahorro de tiempo y recursos.

En el caso particular de los proyectos de investigación académicos, podría no ser viable para la organización ofrecer una retribución monetaria por la ayuda prestada por

[4] https://www.zooniverse.org

los usuarios en la identificación de puntos de interés, debido a la potencial masividad de participantes y a los recursos económicos limitados. Debido a esto, es posible optar por incluir elementos de *Gamification* [Werbach2014] en los proyectos, para que de ese modo reciban una recompensa o contraprestación por su colaboración en forma de entretenimiento, prestigio y/o logros [Hamari2011] dentro de la plataforma.

Debido además a que es posible aplicar el mismo esquema de localización de puntos en diversas aplicaciones o juegos online para dar colaboración en múltiples áreas científicas, sería viable generalizar todas estas características y mecánicas en un framework que pueda ser personalizado y configurado para su adaptación a una materia en particular por un desarrollador de software.

1.2. Objetivos

El objetivo principal de este trabajo es definir, implementar y documentar un framework que le permita a un desarrollador, de manera simple e intuitiva, crear instancias de aplicaciones web orientadas a la identificación, mediante Google Maps y Google Street View, de puntos de interés para una determinada materia, que luego serán puestos a disposición de expertos del área para llevar a cabo sus proyectos de investigación.

Con el desarrollo de este framework se espera que las aplicaciones resultantes cuenten por defecto con elementos de interacción social y de Gamification que las hagan atractivas para potenciales usuarios o jugadores y que de ese modo se vea incrementada su participación y colaboración [Bowser2013].

1.3. Contribuciones

Se espera poder brindar una herramienta que sea de utilidad para las áreas de investigación de las diversas Unidades Académicas de la Universidad Nacional de La Plata, que les permita obtener mejores y más rápidos resultados en la recolección de fotografías geolocalizadas para la realización de sus experimentos.

Una vez finalizado el desarrollo, se pretende liberar el mismo como software de código abierto mediante la plataforma *GitHub*[5], para que pueda ser utilizado no sólo en

[5] https://github.com/

el ámbito de la UNLP sino por cualquier otra persona o entidad a la que pudiera serle de utilidad un framework con las características antes mencionadas.

1.4. Organización del documento

A continuación, se hace una breve descripción de cada uno de los capítulos que componen la tesina.

En el Capítulo 2, *Ciencia Ciudadana Fotografiando la Vía Pública*, se desarrollan conceptos generales sobre Ciencia Ciudadana y particularmente los casos concretos que motivaron el desarrollo de este trabajo. También se hace un repaso de los proyectos de los que está compuesto Cientópolis y que guardan relación con el framework a desarrollar y los requerimientos básicos que dicho framework debería cumplir.

El Capítulo 3, *Trabajos Relacionados*, se centra en otros proyectos que se sirven de datos de geolocalización para ayudar a la investigación sobre diversos temas. De cada uno de estos proyectos se destacan sus características, puntos fuertes y cuestiones mejorables para intentar trasladarlas a Spotters, el framework a desarrollar.

El Capítulo 4, *Estrategia*, detalla las características principales de Spotters como framework, los puntos de extensión que debería tener (HotSpots) y aquellas cuestiones que son fijas o que requerirían modificaciones más profundas sobre el framework (FrozenSpots). También se hace referencia al proyecto de Cientópolis conocido como Metajuego, el cual se encarga de englobar las características de Gamification presentes en Spotters y en otros proyectos de la plataforma.

En el Capítulo 5, *Recorrido por Spotters*, se hace una demostración de cómo está conformado Spotters, sus características y secciones principales, junto con muestras de sus pantallas.

En el Capítulo 6, *Arquitectura del Framework*, se desarrollan las principales decisiones de diseño que se tomaron al momento de plantear Spotters. Además, se explica en detalle el conjunto de clases que componen a Spotters y sus interacciones tanto internas como con otros sistemas. Finalmente se presentan con mayor nivel de detalle los puntos de extensión para desarrolladores o HotSpots.

En el Capítulo 7, *Plataforma de Instanciación*, se hace una descripción de los pasos necesarios para instanciar Spotters partiendo del código fuente y realizando las configuraciones necesarias para cada componente. También se explican en detalle cada una de las funcionalidades a las que puede acceder cada usuario, de acuerdo a su rol.

El Capítulo 8, *Implementación*, contiene la descripción técnica de cada uno de los componentes de Spotters, lenguajes de programación utilizados, librerías, frameworks, APIs y sistemas de base de datos.

El Capítulo 9, *Caso de Estudio: Bellas Artes*, detalla las consideraciones que se tomaron al momento de crear la instancia particular de acuerdo a los requerimientos que surgieron de las reuniones con investigadores del proyecto de investigación de la Facultad de Bellas Artes.

En el Capítulo 10, *Conclusiones*, se realiza un análisis de todo el proceso de realización del presente trabajo y se expresan los resultados y conclusiones obtenidas de la implementación y utilización del framework y su desarrollo.

Finalmente, el Capítulo 11, *Trabajos Futuros*, detalla las posibles extensiones que podrían hacerse sobre Spotters a futuro, para mejorar puntos que quedaron fuera del alcance del presente trabajo.

2. Ciencia Ciudadana Fotografiando la Vía Pública

2.1. Ciencia Ciudadana

Es posible definir a la Ciencia Ciudadana como una técnica de investigación que permite que el público en general participe en la búsqueda de información científica [Bhattacharjee2005]. De esta manera, la Ciencia Ciudadana trata sobre cómo personas comunes pueden realizar un aporte a la ciencia sin poseer la totalidad de los conocimientos técnicos que posee un experto en una determinada materia.

La Ciencia Ciudadana es posible gracias a la voluntad de colaborar de las personas y al deseo que poseen de involucrarse en tareas de índole científica. De este modo, estos individuos pueden brindar datos de muestra a los profesionales o realizar tareas repetitivas, pero de fácil aprendizaje. Así, el científico tiene la posibilidad de completar tareas que de otra manera no serían viables, y las personas, como contraprestación, adquieren nuevos conocimientos y pueden sentirse motivadas porque son parte de la comunidad científica.

2.2. Tecnologías para la Ciencia Ciudadana

Al momento de escribir este trabajo, el universo de herramientas disponibles para que la comunidad científica genere aplicaciones de Ciencia Ciudadana es relativamente acotado. En el capítulo anterior se mencionó la herramienta de construcción de proyectos provista por Zooniverse. Además de esta herramienta, Zooniverse también puso a disposición de la comunidad el proyecto *Panoptes*[6] [Bowyer2015], que permite a los científicos crear una instancia propia de Zooniverse, en un servidor propio y con sus personalizaciones.

En ambos casos, tanto el constructor de Zooniverse como Panoptes, por defecto proveen funcionalidad para la clasificación de muestras, pero no poseen funcionalidad que le permita a los voluntarios tomar y subir sus propias muestras a la plataforma.

2.3. Gamification en la Ciencia Ciudadana

Uno de los inconvenientes que debe afrontar la Ciencia Ciudadana a diario, es el de encontrar la motivación que debe darse al público para que colabore en las tareas

[6] https://github.com/zooniverse/Panoptes

que deben completarse para una actividad en particular. En otro ámbito, esta motivación podría ser de tipo económica, sin embargo, los proyectos científicos no suelen contar con el capital económico suficiente para poder pagar por el tiempo que le tomaría a un conjunto numeroso de personas el realizar las pruebas o tareas. Por otro lado, si la tarea no resulta motivadora o gratificante, es muy probable que sea abandonada, o que no se realice a conciencia. Esto, podría provocar la introducción de errores y generar resultados incorrectos debido a un mal desempeño de las personas que completan las tareas, lo que podría además llegar a impactar negativamente en el resultado de los proyectos de investigación.

Una forma de captar usuarios y hacer masiva a la Ciencia Ciudadana que ha probado ser exitosa es el uso de técnicas de Gamification [Bowser2013] [Deterding2011], es decir, convertir las tareas que necesitan completar los científicos en tareas lúdicas. De esta manera, se puede lograr que una persona se divierta y esté satisfecha al mismo tiempo que ayuda a la ciencia.

La incorporación de Gamification en un proyecto de Ciencia Ciudadana, permite que el hecho de completar una tarea científica pase a un segundo plano, y que los voluntarios, de forma inconsciente, mientras se divierten, compiten por distintos logros y avanzan de niveles, realizan tareas que brindan datos de utilidad para los científicos. Los participantes también obtienen una recompensa extra al intentar completar niveles, obtener medallas, juntar experiencia, entre otras características posibles. Además, si se logra que el juego sea interesante, puede volverse masivo, atrayendo cada vez a más gente para que participe del mismo [Denny2013].

Un caso renombrado del uso de Gamification en un proyecto de Ciencia Ciudadana es el de *FoldIt*[7], donde todo el proceso de contribución fue convertido en un juego. El objetivo es que los jugadores contribuyan al trabajo sobre la identificación de la forma de ciertas proteínas. Con esta información aportada por los jugadores, se espera lograr avances en la lucha contra el cáncer, el HIV y el Alzheimer.

[7] https://fold.it/portal/

2.4. Toma de Muestras Fotográficas en la Vía Pública

Uno de los grandes aportes que pueden realizar los participantes, que no necesariamente son científicos, es la toma de muestras fotográficas en la vía pública, es decir, tomar fotografías que podrían ser de interés para el científico por su contenido, ayudándolo a encontrar casos de estudio particulares. Sin este aporte, el investigador debería recorrer él mismo las áreas y tomar las fotografías o contratar personas para que realicen dicha tarea. En ciertos casos, cuando el área de interés es muy extensa, sin la colaboración de los voluntarios sería materialmente imposible de abarcar para el científico.

Otra de los aportes que puede hacer un ciudadano es la catalogación de imágenes que ya fueron capturadas con anterioridad. Por ejemplo, puede ser que el científico ya posea las imágenes y solamente sea necesario realizar su clasificación. De esta manera, si al ciudadano se le explica cómo catalogar dichas imágenes (qué aspectos de la imagen debe tener en cuenta, qué datos son de interés para el proyecto, qué particularidades debe observar, etc.), sería posible que realice esta tarea con un porcentaje de eficacia muy similar al del profesional, y que ese porcentaje de eficacia aumente en la medida en que realice más clasificaciones.

Estas dos formas de aportar a un proyecto de Ciencia Ciudadana, presentan cada una sus propios desafíos. En lo que respecta al primer caso, si las fotografías son obtenidas con anterioridad, los casos de estudio se limitan a un subconjunto de los posibles casos que existen en la realidad, y la potencial ayuda que podrían prestar los voluntarios se ve limitada. Respecto del segundo caso, si las fotografías son tomadas de la vía pública, sólo pueden colaborar los ciudadanos con un Smartphone con acceso a internet y que habiten en las cercanías del área de interés, lo que limita en gran parte la cantidad de personas que pueden ayudar en la captura de imágenes. Además, es necesario que la persona se mueva por distintos lugares para obtener una mayor variedad de fotografías, y se requiere un grupo mayor de personas geográficamente dispersas para poder abarcar varias zonas del área de interés.

2.4.1. Escenario de Uso: Facultad de Arquitectura

El primer caso de estudio que inspira este trabajo surge desde el área de Arquitectura, donde para un proyecto de investigación se necesita identificar la presencia de vados peatonales en la ciudad de La Plata, para así tener un mejor panorama de la situación de accesibilidad para personas con movilidad reducida en la vía pública.

El principal inconveniente que tiene el proyecto es que no se poseen datos concretos en la ciudad de La Plata respecto de donde se encuentran ubicados estos vados peatonales, de forma que una persona en sillas de ruedas no puede saber qué camino tomar si necesita ir de un lugar a otro.

Para la realización del proyecto, no se poseen las fotografías de la ciudad de La Plata para poder marcar las rampas existentes, y, además, es muy costoso realizar esta tarea para el proyecto por medio de voluntarios que recorran la ciudad. Por lo tanto, se intentó buscar una alternativa para solucionar este inconveniente. En este punto surge como mejor opción la herramienta Google Street View, que permite visualizar toda la ciudad desde la perspectiva de un automóvil, con vista de 360º, también llamada vista panorámica.

2.4.2. Escenario de Uso: Facultad de Bellas Artes

El segundo proyecto que motiva este trabajo surge de parte de Bellas Artes. La necesidad concreta radica en encontrar esculturas y monumentos dispersos por la ciudad, desde monumentos y próceres hasta posibles grafitis, creados a partir del año 2000.

Al igual que el proyecto de Arquitectura, no se posee una base de datos con fotografías de la ciudad para poder catalogar las distintas obras de arte que se encuentran, por lo que también resultaría de utilidad la tecnología provista por Google Street View, ya que dada la naturaleza del proyecto es poco probable que se tenga de antemano una base de fotografías tan extensa para poder catalogarla.

El proyecto de Bellas Artes presenta un requerimiento adicional, ya que es necesario explicarle al usuario cómo debe catalogar las esculturas o posibles grafitis que encuentran distribuidos por la ciudad. De esta manera, se hace necesario el diseño de un

tutorial que pueda ser visto por el usuario cada vez que lo desee, enseñándole como se realiza la clasificación en función de los elementos que se ven en la vista de 360º.

Este proyecto se aborda en más detalle en el Capítulo 9, donde se lo desarrolla como un caso de estudio.

2.5. Limitaciones y Dificultades

En la Ciencia Ciudadana podemos encontrar diversas limitaciones y dificultades, algunas vinculadas a la información con la que es necesario contar antes de presentar el proyecto a los voluntarios, otras asociadas con la tecnología o herramientas a las que deben tener acceso dichos voluntarios para poder colaborar con el proyecto, y, finalmente, otras en lo que respecta a las distancias que deben recorrer físicamente los voluntarios para acceder a las áreas de interés para el proyecto.

Para el caso de aquellos proyectos que consisten en la clasificación de imágenes sobre las que se deben identificar ciertos rasgos o características, una de las principales dificultades es poder contar con las imágenes a analizar, previo a la creación del proyecto. Es decir que el científico debe encargarse de conseguir el conjunto de fotografías necesarias para el análisis. Esta tarea muchas veces resulta poco práctica, ya que se trata de conjuntos de cientos o miles de imágenes y de una calidad acorde a lo que se desee identificar en cada una. Dichas imágenes pueden haberse obtenido previamente a partir de otro proyecto o el científico puede haberse ocupado de capturarlas él mismo o por medio de otros voluntarios.

Para el caso donde son los ciudadanos quienes deben obtener las imágenes a analizar, y donde el científico solamente propone el conjunto de requerimientos que necesita obtener para la investigación, el principal inconveniente que se presenta es que es necesario que quienes colaboren con la investigación cuenten con Smartphones con algunas características especiales, como por ejemplo, cámara fotográfica, GPS y una conexión a internet para poder enviar los datos recolectados a los servidores para luego ser analizados.

Otra dificultad que surge en la toma de imágenes por parte de los ciudadanos es la dificultad de conseguir personas que se encuentren geográficamente dispersos para

poder abarcar un área mayor y así cubrir el área requerida por el proyecto de investigación.

Por lo tanto, si bien contar con la ayuda de los colaboradores para obtener las imágenes puede resultar útil, suma dos complicaciones adicionales para el científico. Por un lado, la cantidad de información que debe ser analizada puede llegar a ser mucho mayor, y hay que tener alguna forma de filtrar contenido que no es válido, y por el otro, es necesario enseñarles a los colaboradores cómo tomar dichas imágenes para que sean de utilidad a la hora de ser analizadas.

Finalmente, una dificultad que se encuentra presente en todos los proyectos de Ciencia Ciudadana, tengan o no que ver con tareas referidas a información geográfica, consiste en obtener voluntarios dispuestos a colaborar y mantener el interés de los mismos en el proyecto, dado que, si las tareas que deben realizar les resultan aburridas o poco gratificantes, se verá reducido no sólo el número de voluntarios, sino también la calidad de los datos que aporten.

2.6. Objetivos Específicos: Requerimientos

A partir de las reuniones que se mantuvieron con representantes de los proyectos de Arquitectura y Bellas Artes, y en base a los conceptos mencionados anteriormente de Ciencia Ciudadana y Crowdsourcing, es que se llega a la conclusión de que es necesario construir una plataforma que permita a los científicos instanciar múltiples proyectos para la toma de muestras de una manera rápida y sencilla, es decir, construir un framework para realizar proyectos de Ciencia Ciudadana.

Profundizando un poco más, podemos encontrar los siguientes requerimientos básicos que deberían resolverse o mejorarse respecto de otros similares en esta tesis:

- Navegación de un área geográfica mediante el mapa y la vista panorámica, para así evitar la necesidad de capturar fotografías manualmente por parte de los usuarios.
- Identificar un punto de interés en una vista panorámica, para poder clasificar al mismo y tomarlo como caso concreto en la investigación
- Realizar una clasificación sobre los puntos de interés marcados, para sumar más información a aquello que se observa a simple vista en las imágenes.
- Proveer hilos de discusión sobre cada punto marcado, donde sea identificable la participación del experto en la materia, con el fin de hacer la aplicación

más atractiva a los usuarios y de esta manera mejorar el tiempo de permanencia y calidad de los aportes proporcionados por los colaboradores.

- Tutorial de entrenamiento para nuevos usuarios, con la finalidad de brindarles un entrenamiento que les permita realizar colaboraciones más acertadas de acuerdo a las necesidades del proyecto.
- Votación de clasificaciones y mensajes para simplificar la labor del experto, evitando de ese modo que se pierda tiempo en el análisis de puntos que fueron identificados de forma errónea o malintencionada.
- Utilizar el framework web Spotters para construir la aplicación del escenario de uso planteado por la Facultad de Bellas Artes.
- Publicar Spotters de manera abierta (Licencia *GPL v3*[8] o similar) para que sea utilizado o mejorado libremente por cualquier persona.

[8] https://www.gnu.org/licenses/gpl-3.0.en.html

3. Trabajos Relacionados

En este capítulo se presentan proyectos que han servido de inspiración para la concepción del presente trabajo de tesis y otros casos de éxito de la Ciencia Ciudadana.

3.1. Cientópolis

Cientópolis[9] es un proyecto creado por el LIFIA[10] que agrupa un conjunto de ideas y aplicaciones, algunas ya implementadas y otras en desarrollo, con un perfil orientado hacia la Ciencia Ciudadana y la Ciencia Abierta.

Galaxy Conqueror[11] es un proyecto desarrollado en el marco de Cientópolis y que fue presentado como una Tesina de Grado en la Facultad de Informática de la UNLP [Celasco2015]. Consiste en una aplicación de Ciencia Ciudadana que permite de forma colaborativa identificar posibles galaxias en una fotografía de dimensiones considerables de un cuadrante del cielo, que requeriría un tiempo extenso de dedicación de los expertos del Observatorio para la identificación de cada punto.

Otro de los proyectos que componen Cientópolis es *Collaboratory*, que consiste en una instancia de la plataforma Panoptes del proyecto Zooniverse con modificaciones para ajustarla a las necesidades particulares de Cientópolis, como, por ejemplo, la internacionalización de las vistas para sumar el idioma español y la adecuación de la identidad.

Runaway Stars[12] es otro de los proyectos de Cientópolis que se sirve de cientos de imágenes de posibles *estrellas fugitivas* o *Runaway Stars* que necesitan ser analizadas por voluntarios en busca de características que permitan confirmarlas como tales o descartarlas. Este proyecto es liderado por astrónomos de la Facultad de Ciencias Astronómicas y Geofísicas de la UNLP y se basa en Collaboratory, el servicio de micro tareas de Cientópolis, para distribuir las tareas a los voluntarios y retornar sus opiniones a los científicos, descripto previamente.

[9] https://www.cientopolis.org/
[10] http://www.lifia.info.unlp.edu.ar/lifia/es
[11] https://galaxyconqueror.cientopolis.org/
[12] https://www.cientopolis.org/portfolio/runaway-stars/

El proyecto *Samplers*[13] busca construir un framework para desarrollar aplicaciones móviles de toma de información (muestreo/sampling) para proyectos de Ciencia Ciudadana. En principio abarca aplicaciones que tomen imágenes, videos y audio geoposicionados. A futuro podría incorporar otro tipo de sensores de los dispositivos de captura.

Finalmente, el proyecto del Metajuego o Metagame de Cientópolis, consiste en una forma de centralizar los elementos de Gamification presentes en cada uno de los proyectos antes mencionados, incluido Spotters, con el objetivo de que todos ofrezcan una capa de Gamification consistente entre sí, sin la necesidad de que cada uno lo implemente por su cuenta, y permitiendo además integrar la colaboración en varios proyectos para otorgar logros globales.

3.2. AppEar

El proyecto *AppEar*[14] involucra la Ciencia Ciudadana con el medio ambiente. Su principal objetivo es concientizar a las personas para que cuiden y aprendan sobre el medio ambiente que los rodea a la vez que se obtienen datos críticos de su estado a través del uso de la aplicación.

AppEar funciona de la siguiente manera:

- Se descarga la aplicación al celular.
- El usuario debe ubicarse en el sitio que desea evaluar, como, por ejemplo, la ribera de un río, un lago o un estuario.
- Se deben responder ciertas preguntas de interés, como si hay ganado cerca, cómo es la vegetación, si hay plantas acuáticas, entre otras.
- Se deben tomar fotografías del lugar que se está observando.
- Se envían las fotografías para completar el proceso.
- Se visualiza en el mapa la información enviada y el puntaje obtenido.

Como puede destacarse, AppEar involucra muchos aspectos que son de utilidad, como el workflow de preguntas o ubicar un punto de interés en un lugar específico del mapa. Por otro lado, posee algunas limitaciones en cuanto a la gente que lo pueda utilizar.

[13] https://www.cientopolis.org/portfolio/samplers/
[14] http://www.app-ear.com.ar

Una de las limitaciones más grandes que se tiene es que la persona debe tener un celular para poder utilizar la aplicación, además, este dispositivo debe poseer GPS para la geolocalización, cámara fotográfica para poder sacar las fotos y una conexión de datos de internet para poder completar el proceso. Esto hace que la cantidad de gente que pueda contribuir a la Ciencia Ciudadana disminuya considerablemente. Otra de las complicaciones que se encuentran en AppEar es que la persona debe moverse físicamente hasta el lugar que desea analizar, lo que dificulta mucho la tarea de aquellas personas que quiere colaborar con el proyecto, pero que no tienen forma de movilizarse hasta las cercanías de un río.

Un punto muy logrado en la aplicación de AppEar es el de Gamification, ya que le proporciona al usuario puntaje, niveles y reconocimientos por los datos enviados. En la página principal del sitio puede visualizarse una tabla de clasificaciones con los usuarios que han realizado mayores aportes a la comunidad, haciendo que exista una especie de competencia entre los mismos, fortaleciendo el interés de seguir aportando por el simple hecho de competir contra otros usuarios.

3.3. Mosquito Alert

Esta plataforma utiliza la Ciencia Ciudadana para investigar y controlar mosquitos transmisores de enfermedades globales; sobre todo, en este último tiempo, para combatir al Dengue, el Chikungunya y el Zika.

Mosquito Alert[15], al igual que AppEar, utiliza una aplicación para dispositivos celulares que permite interactuar con la plataforma. Una vez que la aplicación se encuentra instalada en el celular, podemos tomar fotografías de alertas de mosquitos o lugares de crianza de los mismos.

Estas fotografías son catalogadas por expertos antes de hacerlas públicas en el mapa, y el principal objetivo es que los científicos puedan analizar la expansión del mosquito en distintas regiones, pudiendo tomar medidas de seguridad en zonas que se creen que pueden ser focos de infecciones debido a los casos de avistamiento del mosquito.

[15] http://www.mosquitoalert.com

De este proyecto se tomaron dos ideas de suma practicidad a la hora de desarrollar un framework sobre Ciencia Ciudadana. Por un lado, en la página web se ven múltiples tutoriales que explican cómo debería verse el mosquito, cómo tomar correctamente las fotos para que los expertos analicen las mismas, cómo atrapar y matar un mosquito, etc. Además del uso de tutoriales, Mosquito Alert permite que los usuarios voten y clasifiquen las fotografías que realizan otros usuarios de la comunidad, de manera que los expertos reciben la información ya procesada y solamente les queda por validar aquellas fotografías que son ambiguas o en las que no fue posible llegar a una conclusión acertada por parte de la comunidad.

El punto negativo, al igual que la aplicación de AppEar, es que el usuario necesita un dispositivo móvil para poder instalar la aplicación, además de cámara fotográfica y conexión a internet para poder enviar los datos.

3.4. The Plastic Tide

Este proyecto fue incorporado recientemente a Zooniverse y consiste en que los voluntarios analicen imágenes de las costas tomadas mediante drones, con el fin de localizar residuos plásticos alojados en las mismas.

Lo novedoso de *The Plastic Tide*[16] es que se busca que mediante la información aportada por los colaboradores, se entrene un algoritmo de *Machine Learning*, para que posteriormente las fotos puedan ser analizadas de forma automática, sin intervención humana.

Adicionalmente, todo el trabajo realizado por los usuarios en la clasificación de imágenes, las imágenes capturadas por los drones y los algoritmos utilizados, serán puestos a disposición de la comunidad para que puedan ser utilizados por autoridades o proyectos de investigación de distintos países para monitorizar sus propias costas.

3.5. Conclusiones

Las tres aplicaciones analizadas tienen objetivos en común, la utilización de la Ciencia Ciudadana para aportar datos e información necesaria a científicos en un tema en particular. Sin embargo, específicamente AppEar y Mosquito Alert, tienen

[16] https://www.zooniverse.org/projects/theplastictide/the-plastic-tide/classify

inconvenientes en común: la obligatoriedad de utilizar dispositivos tecnológicos relativamente sofisticados para poder utilizar sus plataformas.

Con el desarrollo de este trabajo se busca solucionar estos aspectos utilizando la tecnología provista por Google, además de estar desarrollada como una aplicación web, para que pueda ser vista de manera correcta desde cualquier dispositivo, ya sea una computadora de escritorio o un dispositivo móvil.

Otra característica que poseen las aplicaciones mencionadas con anterioridad, es que únicamente pueden utilizarse para el objetivo por el cual fueron creadas, y si una persona tiene una idea similar que puede llegar a aportar valor a la comunidad, no tiene forma de implementarla a menos que contrate un grupo de desarrolladores y financie su construcción, es decir, que ninguna de estas aplicaciones es generalizable.

4. Estrategia

En este capítulo se desarrolla la estrategia general seguida en este trabajo: la construcción de un framework orientado a aplicaciones de Ciencia Ciudadana. Se discuten las decisiones que se tomaron durante el planteo inicial y sus características principales.

4.1. Propuesta General

Tal como fue mencionado previamente, se mantuvieron un conjunto de reuniones con profesionales e investigadores de ciertas áreas académicas de la Universidad Nacional de La Plata, a partir de las cuales surgen una serie de requerimientos. A partir de estos requerimientos surge la idea de crear un framework que permita generalizar las características de las aplicaciones necesarias para aportar datos a cada proyecto y permitir además crear nuevas según surja la demanda. Se decidió denominar a este framework como Spotters.

De las necesidades y proyectos planteados en esas reuniones, surgió un patrón similar al que implementa Galaxy Conqueror, el proyecto ya mencionado en el Capítulo 2, pero no para la identificación de galaxias en una fotografía de grandes dimensiones (con las dificultades que conlleva para los científicos encontrar y cargar estas imágenes de gran tamaño), sino para la identificación de puntos con ciertas características particulares en un área geográfica. Aprovechando para ese objetivo las funcionalidades cartográficas que proveen ciertas herramientas online, pudiendo utilizar sus mapas y recursos de manera gratuita.

De los proyectos de Bellas Artes y Arquitectura, ya analizados en el Capítulo 2, surge que las tareas podían ser realizadas por personas sin conocimientos específicos de cada una de las áreas, siendo entrenados previamente con un tutorial o con simples ejemplos de cómo o qué deben buscar en la aplicación. Esto hace que ambas aplicaciones sean idóneas para un proyecto de Crowdsourcing [Brabham2008].

En una primera instancia, para la realización de estos proyectos, se pensó en una aplicación que automatice la búsqueda de puntos de interés mediante el análisis de las imágenes geográficas provistas por las plataformas de mapas online. Sin embargo, debido a la magnitud del terreno a cubrir (como mínimo, toda la ciudad de La Plata) y a

la dificultad de crear un sistema que sea capaz de identificar acertadamente un punto de interés de diversas naturalezas y que dependen en muchos casos de la subjetividad de quien observa, se descartó esta posibilidad durante la etapa de análisis de los requerimientos.

Con estos dos proyectos en mente, se pensó la forma de generalizar una solución que permitiera abordar ambos escenarios, donde fuera posible que un conjunto de usuarios realice la identificación de puntos de interés (obras de arte, vados peatonales, etc.) y los clasifiquen mediante un cuestionario definido previamente por los expertos en la materia. Además, para simplificar la labor de los expertos del dominio y ya contar con un cierto grado de validez sobre los puntos de interés marcados, los usuarios podrían emitir una valoración sobre los puntos indicados por otros usuarios, evitando así que varias personas identifiquen el mismo punto, sino que solamente deban emitir una valoración en el caso de que ya se encuentre identificado. Finalmente, como requerimiento principal, se halla también la capacidad del experto en el dominio de calificar los puntos de interés marcados por los usuarios, aceptando o rechazando dichos puntos, de acuerdo a si los considera válidos o no.

Con estos requerimientos en mente, surge la necesidad de crear un framework que permita no sólo implementar soluciones para los dos escenarios antes planteados, sino también poder abordar otros casos similares con relativa facilidad y con poca o ninguna adaptación.

A partir de esto, se toman como base las ideas y el desarrollo de Galaxy Conqueror y comienza un proceso de extrapolación de los conceptos que allí se aplican a nuestro caso particular. Tal como sucede con dicho proyecto, una de las cuestiones más complejas a resolver en Spotters es la de cómo lograr que el usuario se comprometa y continúe participando cuando no hay una contraprestación económica por el tiempo dedicado.

Otro punto que surge en el análisis de los requerimientos, es la posibilidad de generar el interés de la comunidad a través un debate abierto sobre cada punto de interés, en el que puedan participar los científicos para darle más relevancia y prestigio a lo que se está discutiendo. De este modo, Spotters empieza a alejarse de la idea original de Gamification, delegando esta característica a un proyecto que ya se

mencionó con anterioridad en el Capítulo 2 y que es conocido como el Metajuego de Cientópolis, con el fin de proveer una experiencia de ludificación uniforme para toda la plataforma. Con esto en mente, se toma la decisión de priorizar la presencia de elementos sociales en el framework, que promuevan la interacción entre los usuarios.

De este modo, se suma la posibilidad de marcar puntos de interés y votar por ellos, la capacidad de tener hilos de discusión por cada punto, donde los usuarios puedan dejar sus mensajes y consideraciones, y puedan recibir respuestas de los expertos a sus inquietudes. De este modo, si la comunidad que se forma en torno a una aplicación es lo suficientemente activa, tener el feedback de un experto en la materia dentro de un hilo de discusión podría ser una recompensa más importante que un puntaje simbólico o una medalla por realizar una acción.

A grandes rasgos, se busca crear una plataforma/framework generador de aplicaciones de Ciencia Ciudadana, adaptable a casos concretos y que permita realizar las siguientes tareas:

- Navegar un mapa de una región en particular con una vista lo más detallada posible, como la brindada por Google Maps con Google Street View, y permitir el marcado de puntos y la toma de fotos mientras se realiza dicha navegación.
- Ofrecer formas de debate entre los usuarios y expertos para motivar la labor de búsqueda y clasificación.
- Armar un tutorial de capacitación específico para cada instancia de aplicación que entrene a los usuarios para lograr niveles aceptables de identificación de puntos.

4.2. El Framework: FrozenSpots y HotSpots

Podemos definir un framework como una aplicación que por sí misma no tiene una funcionalidad concreta y no se encuentra finalizada, pero que fue pensada y desarrollada para que pueda ser ampliada fácilmente, y de esta manera, resuelva problemas determinados de un dominio específico; es decir, poder construir múltiples aplicaciones de un dominio específico a partir de una aplicación ya existente que sirve como punto de inicio, y que ya resuelve aspectos y requerimientos comunes al dominio donde se está trabajando.

El framework está compuesto por dos conceptos muy importantes que definen qué aspectos pueden editarse del framework (HotSpots) y qué aspectos son estáticos y por lo tanto el programador no podría modificar, ya que sirven de base común a todas las aplicaciones (FrozenSpots).

Dentro de Spotters, los requerimientos que surgieron del análisis, generaron una serie de FrozenSpots y HotSpots, de los cuales se extraen los más importantes para detallarse a continuación.

4.2.1. FrozenSpots

Los principales FrozenSpots identificados en Spotters son:

- Utilización de Google Maps para la primera capa de visualización, donde se pueden encontrar los puntos de interés en distintos colores.
- Utilización de Google Street View para tomar la captura de un punto de interés.
- Al capturar un punto de interés, debe responderse un cuestionario.
- Las clasificaciones pueden votarse de manera positiva o negativa.
- El inicio de sesión se realiza a través de una red social.

Como puede verse, estos FrozenSpots están definidos de acuerdo a la tecnología subyacente sobre la que corre Spotters y las tecnologías de las que se sirve, y si bien para esta versión del framework son FrozenSpots, en un futuro podrían añadirse distintas capas de visualización del mapa y de los puntos de interés, lo que permitiría reutilizar toda la lógica que compete a la clasificación de candidatos, registro e inicio de sesión de usuarios, votación de candidatos, etc., pero aplicado sobre un motor de mapas diferente.

4.2.2. HotSpots

En un framework, es importante que haya la mayor cantidad posible de HotSpots, dado que este es el punto de acceso que poseen los desarrolladores para poder construir una plataforma que resuelva la mayor cantidad posible de requerimientos del problema en cuestión, sin la necesidad de realizar modificaciones en el núcleo del framework. Es por eso, que Spotters cuenta con distintas configuraciones

que permiten realizar múltiples instanciaciones de aplicaciones que tengan como objetivo principal la localización y clasificación de un punto de interés sobre Google Maps y Google Street View.

Entre los principales HotSpots existentes en Spotters se encuentran los siguientes:

- Posibilidad de añadir múltiples redes sociales para realizar el registro e inicio de sesión de un usuario.
- Tutorial interactivo con la posibilidad de editar los textos que se desea que aparezcan en cada uno de sus pasos.
- Workflow de preguntas para la clasificación de un nuevo punto de interés, estas preguntas pueden ser abiertas o cerradas, y tener múltiples caminos posibles en función de las respuestas de los usuarios en la clasificación.
- Marcadores con colores para identificar candidatos ya presentes en el mapa.
- Área donde es posible encontrar candidatos, restringiendo la búsqueda a un cuadrante específico.
- Noticias y alertas para los usuarios al iniciar sesión en la plataforma, para poder comunicar eventos o información de interés.
- Integración con el Metajuego de Cientópolis, es posible indicar a qué instancia se va a enviar la información de interés y de dónde van a obtenerse las medallas obtenidas por los usuarios.
- Inicialización del mapa, se puede configurar el punto específico donde va a estar centrado Google Maps y que nivel de zoom va a tener el cuadrante.

4.3. La Plataforma Como Servicio

Una vez definidos los requerimientos más importantes de Spotters, fue necesario decidir qué tipo de framework iba a implementarse para poder realizar múltiples instanciaciones de aplicaciones de forma fácil y rápida.

Para esto, se decidió implementar un framework que sea extensible por configuración, y que solamente baste con descargarse el código fuente de la aplicación base (el framework) y seguir una serie de pasos para poder instanciar una nueva plataforma.

Para facilitar el uso de la plataforma y para que cualquier persona que esté interesada en realizar un proyecto que se ajuste a las características que ofrece Spotters pueda realizar una aplicación en algunos pasos, se decidió además ofrecer un panel de administración. A través del dashboard que proporciona Spotters, se puede completar todas configuraciones de la nueva plataforma mediante la utilización de formularios web.

La ventaja que provee tener un panel de administración web es que permite garantizar que las configuraciones que se están cargando sean correctas, y que de ese modo no pueda quedar la aplicación en un estado inconsistente o de error. Otro beneficio que trae este sistema es que permite que cualquier persona que tenga una idea, o desee realizar un proyecto con Spotters, pueda bajarse el código fuente desde el repositorio y realizar una instalación completa de su plataforma sin la necesidad de contratar un programador o un experto en informática.

De esta manera, se maximiza el uso de Spotters, facilitando que múltiples personas puedan realizar proyectos que contribuyan a distintas áreas de investigación utilizando Ciencia Ciudadana para resolver sus problemas, o simplemente para tomar datos estadísticos de puntos de interés específicos sobre un área determinada.

4.4. Gamification Como Servicio Externo

Como se mencionó anteriormente, la parte de Gamification se delega a un servicio externo de Cientópolis llamado Metajuego, donde es posible incorporar una capa de juego a todos los proyectos involucrados de una manera transversal. De esta manera, no es necesario que cada una de las aplicaciones diseñe e implemente un juego dentro de un ambiente de Ciencia Ciudadana, sino que solamente con incluir características necesarias para el Metajuego y enviando información de los sucesos acontecidos en la plataforma se puede tener una capa de Gamification consistente a todos los demás proyectos que componen Cientópolis.

Dentro de Spotters, existen dos formas de comunicación con el servicio externo de Gamification, tanto para tomar datos como para alimentar el mismo. Ambos sistemas son explicados a continuación.

Para la alimentación del Metajuego es necesario enviar eventos particulares a un Bus de Eventos, implementado mediante *Apache Kafka*[17], cada vez que el usuario que ha iniciado sesión realiza acciones que se creen de interés para el Metajuego. El Metajuego analiza los distintos eventos que el usuario realiza sobre el conjunto de aplicaciones en las que interactúa y va otorgando medallas de reconocimiento por su aporte a la Ciencia Ciudadana dentro de Cientópolis.

De esta manera, se esperan en el bus de eventos cuatro tipos posibles de interacciones que realiza el usuario:

1. *Login*: Es avisado cada vez que el usuario entra en la aplicación para realizar alguna interacción
2. *Contribution*: Cada vez que el usuario realiza una contribución este evento es disparado. En el caso de Spotters, cada vez que se clasifica un nuevo candidato se le avisa a la capa de Gamification sobre lo sucedido.
3. *Reinforcement*: Son eventos de contribución secundaria, o de apoyo a una contribución ya realizada. En Spotters, cada vez que se realiza una votación sobre un comentario o una clasificación existente este evento es invocado
4. *Social*: Son eventos relacionados con la difusión de Spotters o de Cientópolis en general.

Estos eventos son tratados en una clase especial de Spotters, que se encarga de enviar las peticiones a la instancia de Gamification previamente configurada en la instanciación de la aplicación.

Además de alimentar el bus de eventos del Metajuego, Spotters se encarga de visualizar e informar a los usuarios de las medallas y/o logros que han obtenido por su aporte a la Ciencia Ciudadana a través de Spotters, así como el nivel obtenido dentro del juego.

Toda la integración entre el Metajuego y Spotters se realiza a través de un HotSpot, y se realiza de manera aislada a los demás módulos de Spotters, por lo tanto, es fácilmente personalizable y extensible. Ya que tanto para la obtención de los datos del

[17] https://kafka.apache.org/

usuario, como para la alimentación del Metajuego se realizan peticiones a una API REST de manera asincrónica.

5. Recorrido por Spotters

A continuación, se realiza un recorrido por una instancia de Spotters. Pueden observarse las distintas secciones que la componen y las acciones que pueden realizarse desde cada una.

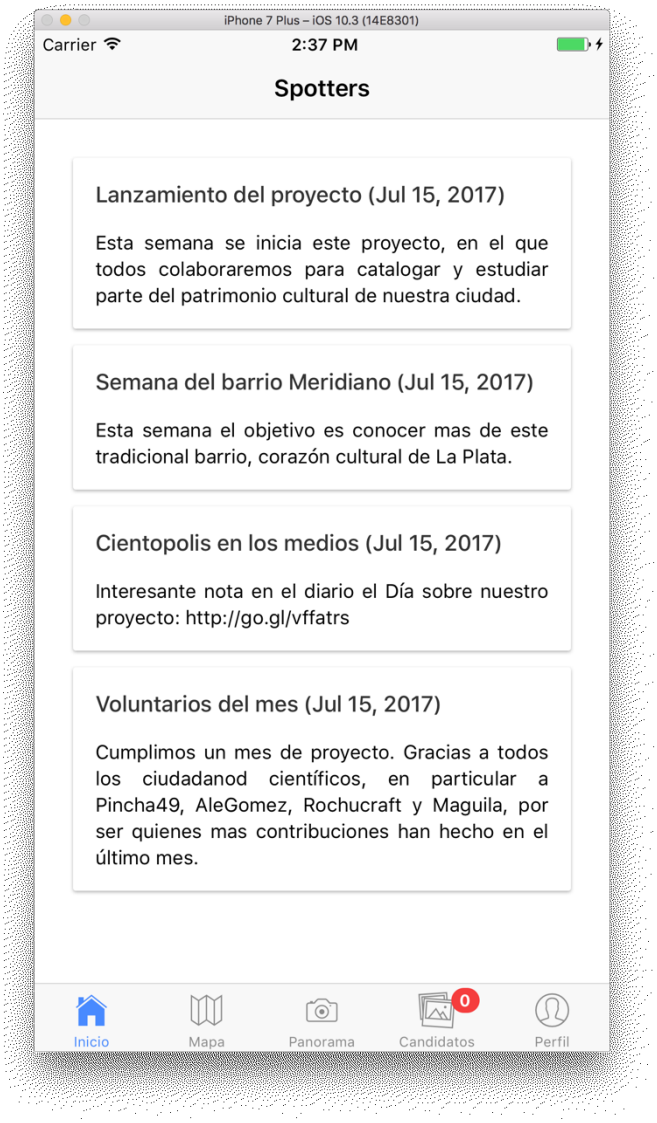

Ilustración 1 - Portada

En la Ilustración 1 puede observarse la página principal o portada de una instancia de Spotters. Esta sección muestra las últimas noticias. En la parte superior se muestra el nombre que fue configurado para la instancia ("Spotters" en este caso particular) y en la parte inferior se muestra una barra de navegación que permite acceder a las demás secciones de la aplicación: Mapa, Panorama, Candidatos y Perfil.

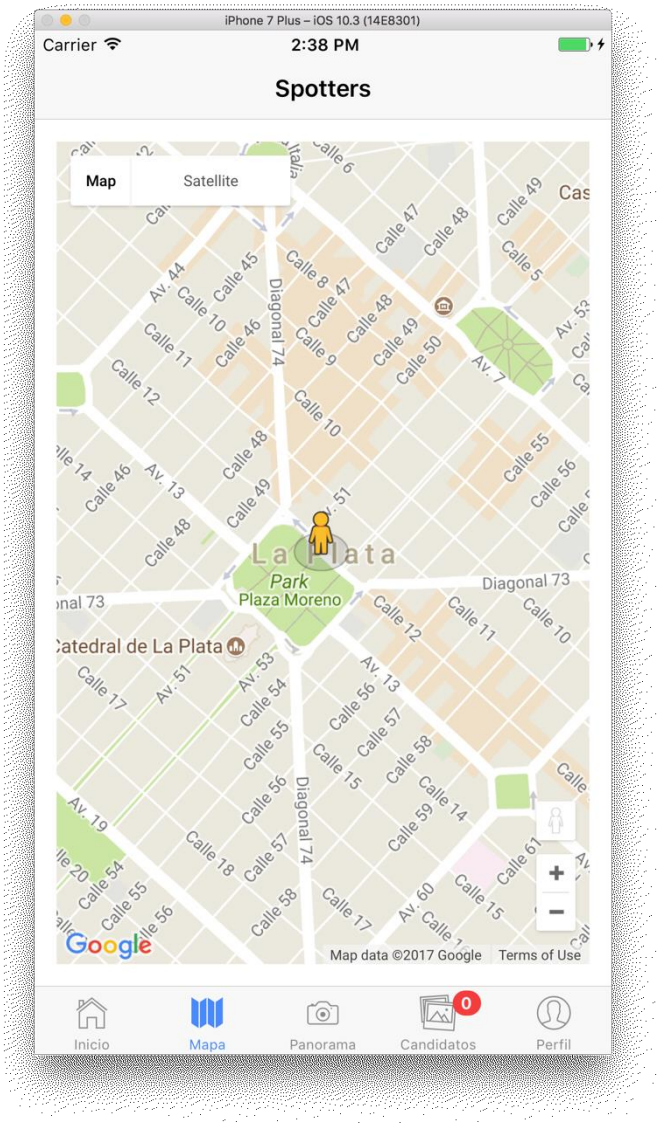

Ilustración 2 - Mapa

Tal como se ve en la Ilustración 2, donde se encuentra activa la vista del mapa, cuya ubicación inicial depende de si el usuario de la aplicación permite el acceso al GPS o no. En caso afirmativo, el mapa estará centrado en las coordenadas indicadas por el dispositivo que esté utilizando. Si el usuario hubiera preferido no brindar esa información, el mapa se centrará en un punto que se encuentra predefinido en la configuración de la instancia.

En el mapa pueden observarse marcas que representan candidatos que ya han sido identificados. El color de la marca representa el estado del candidato. Sólo se cargan en el mapa aquellos candidatos que se encuentran a cierta distancia de donde se

encuentra posicionado el indicador del panorama. Esta distancia es configurable para la instancia.

Desde el mapa es posible trasladar el indicador del panorama que se ve en el centro (figura amarilla) hacia distintos puntos, y el punto en el que se halle esa figura será el punto inicial de la exploración en la vista del Panorama.

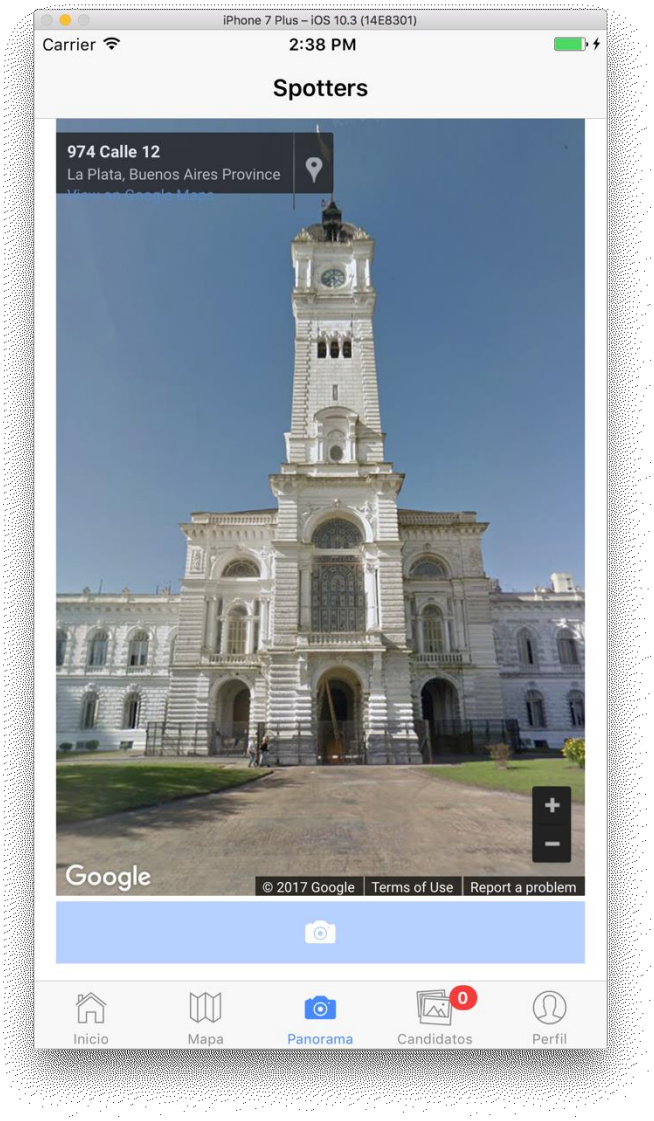

Ilustración 3 - Vista Panorámica

El Panorama o Vista Panorámica (Ilustración 3) permite acceder a una vista de 360º del punto que se indicó desde el mapa, gracias a Google Street View. También permite realizar desplazamientos para recorrer el área. Estos desplazamientos se sincronizan con la vista del mapa, es decir, si el usuario se desplaza dentro del panorama, los cambios se reflejarán también en el mapa.

Desde esta vista es posible también realizar la captura (toma de fotos) de candidatos. En el caso de la Ilustración 3, dicha opción se encuentra deshabilitada debido a que el usuario no ha iniciado sesión, pero se explicará el proceso con más detalle más adelante.

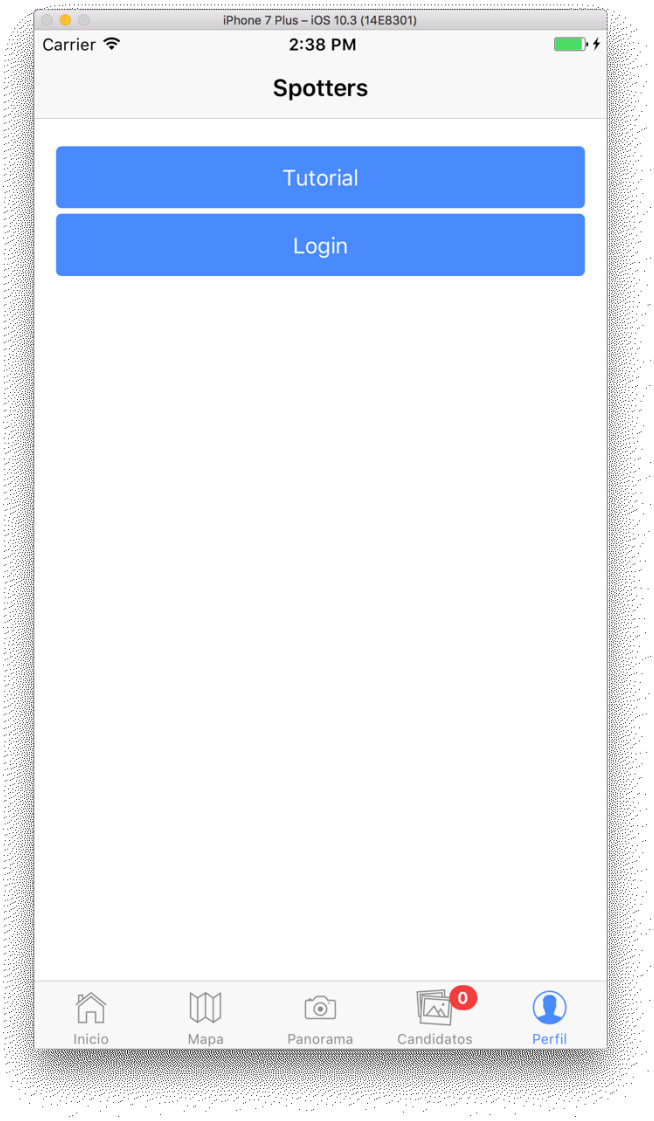

Ilustración 4 - Perfil

La Ilustración 4 muestra la vista del Perfil del usuario cuando todavía no se ha iniciado sesión. Las opciones disponibles permiten acceder al tutorial de entrenamiento del usuario o realizar el inicio de sesión.

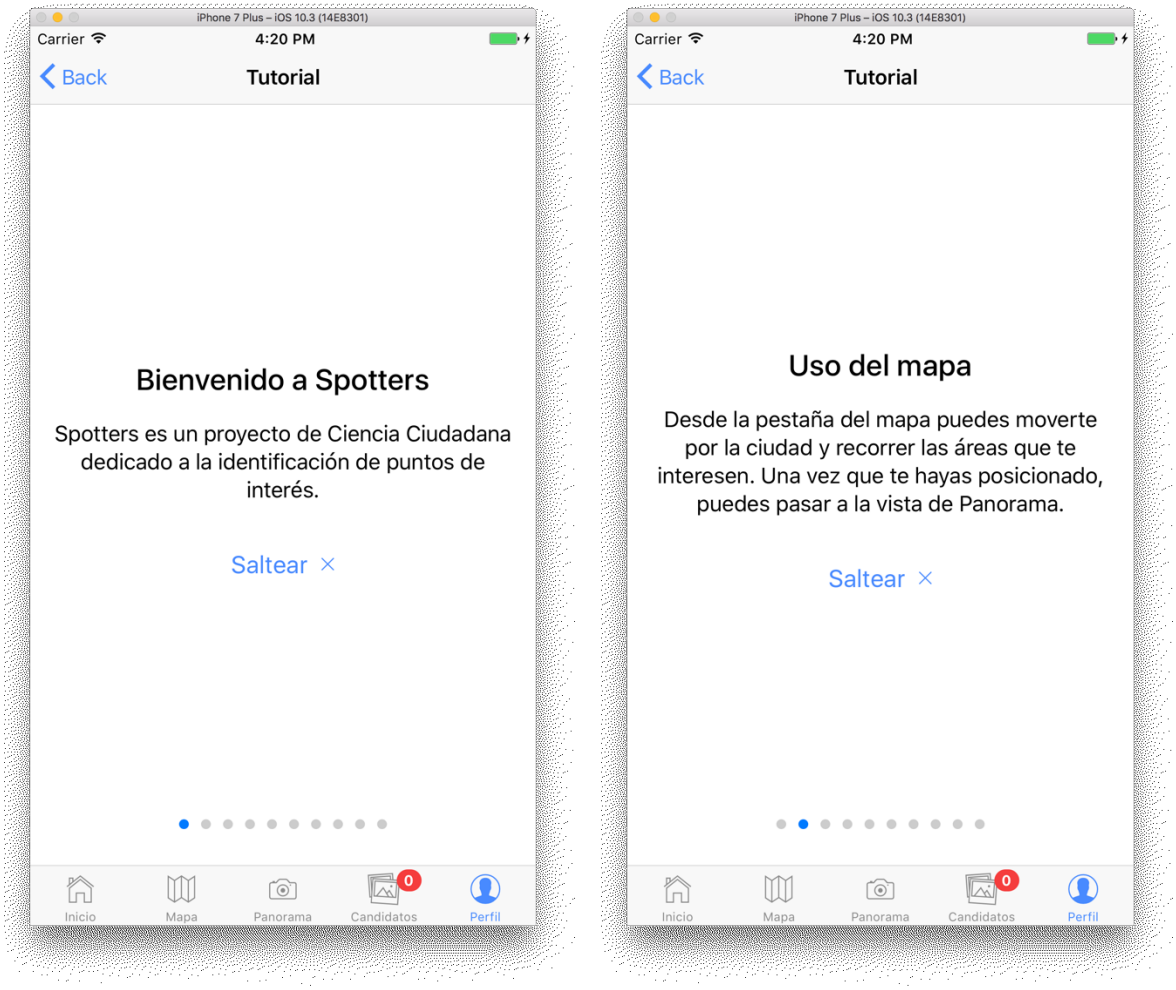

Ilustración 5 - Tutorial

Una vez que el usuario ha iniciado sesión, si es la primera vez que ingresa a la aplicación, se le mostrará el tutorial de entrenamiento (Ilustración 5).

Los pasos del tutorial se configuran al momento de crear la instancia de Spotters, y por cada uno es posible indicar un título y texto.

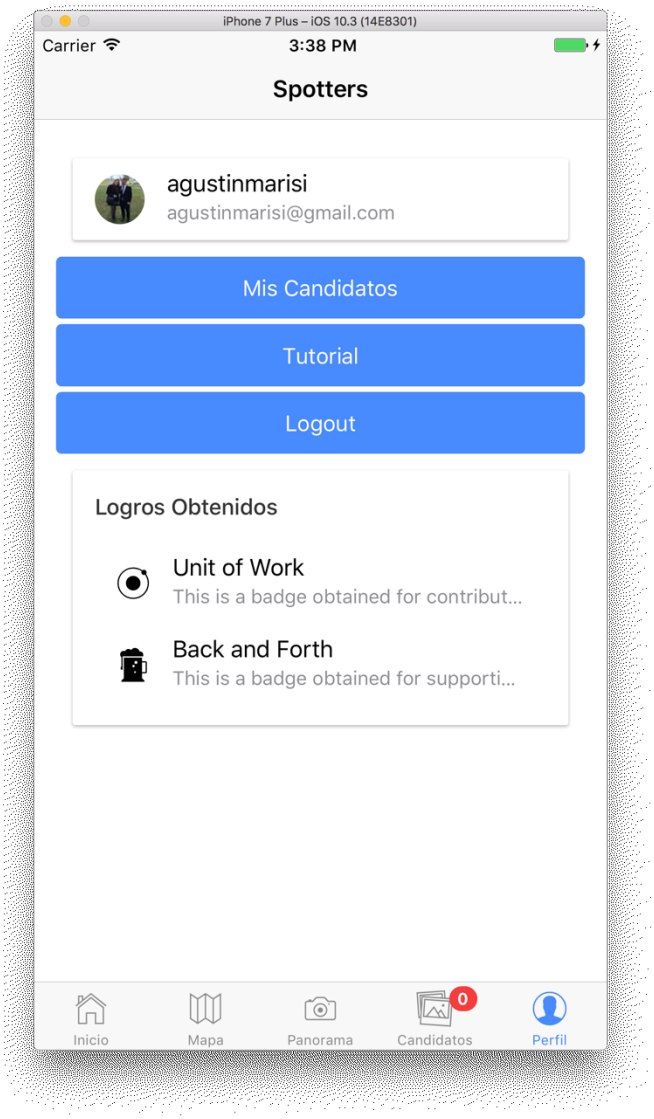

Ilustración 6 - Perfil

En la Ilustración 6 se observa el perfil del usuario una vez que ha iniciado sesión. Desde allí, es posible acceder a la lista de candidatos que ha registrado. También permite observar los logros obtenidos por el usuario para esta instancia de Spotters en particular por medio del Metajuego de Cientópolis.

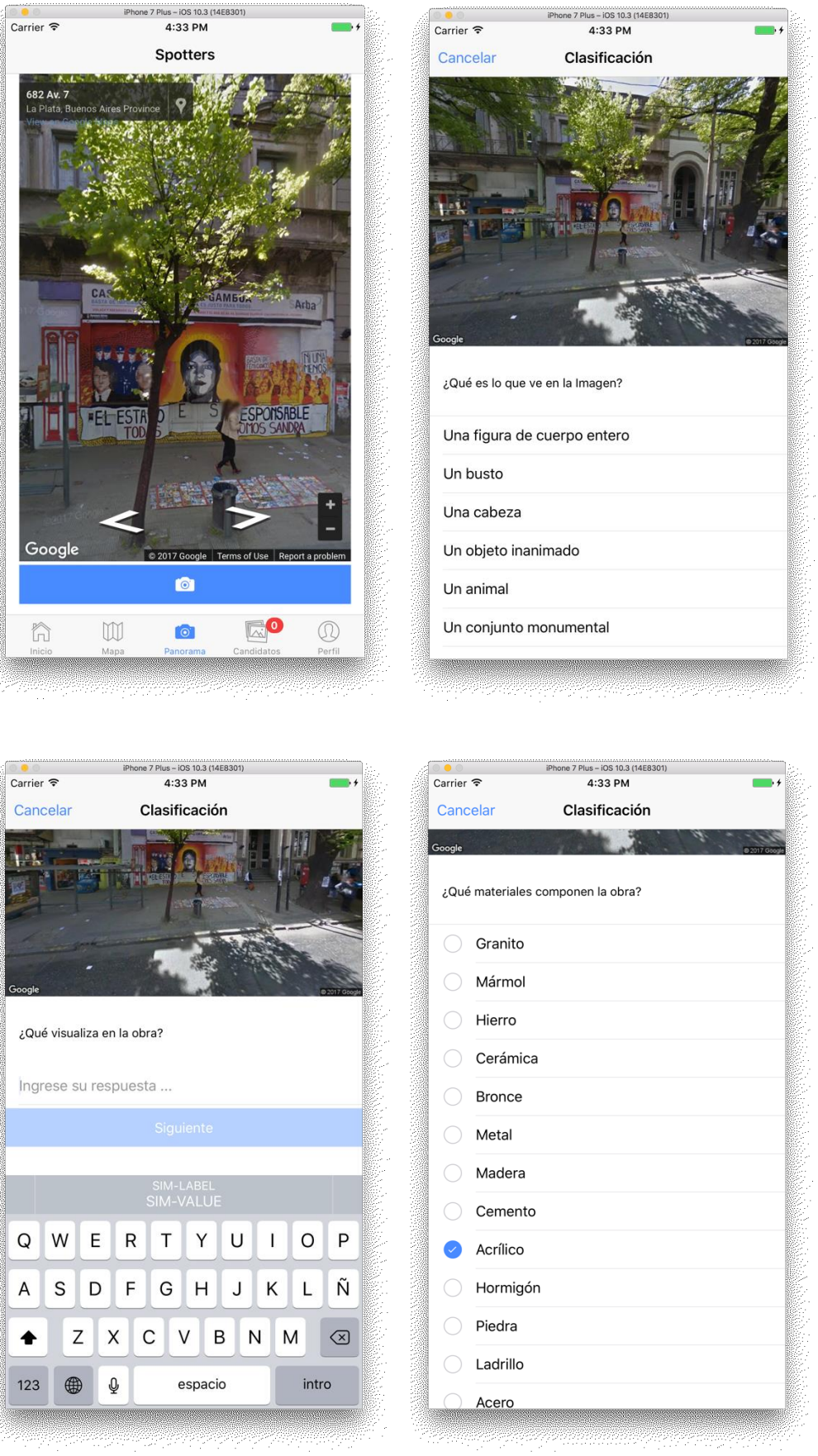

Ilustración 7 - Nuevo Candidato

Al presionar el botón con el ícono de la cámara que se observa en la primera imagen de la Ilustración 7, desde la pestaña Panorama, se accede al cuestionario que se muestra en el resto de las imágenes de la misma ilustración.

Cada una de estas tres imágenes muestra una de las preguntas que componen un cuestionario de clasificación. Una vez respondidas las preguntas, se carga en la aplicación un nuevo candidato con su correspondiente clasificación y fotografía.

Las preguntas que componen el proceso de clasificación, se crean durante el proceso de configuración de la nueva instancia y como puede observarse en las imágenes, existen distintos tipos de preguntas (con opciones múltiples y una respuesta, con opciones múltiples y múltiples respuestas y con texto libre).

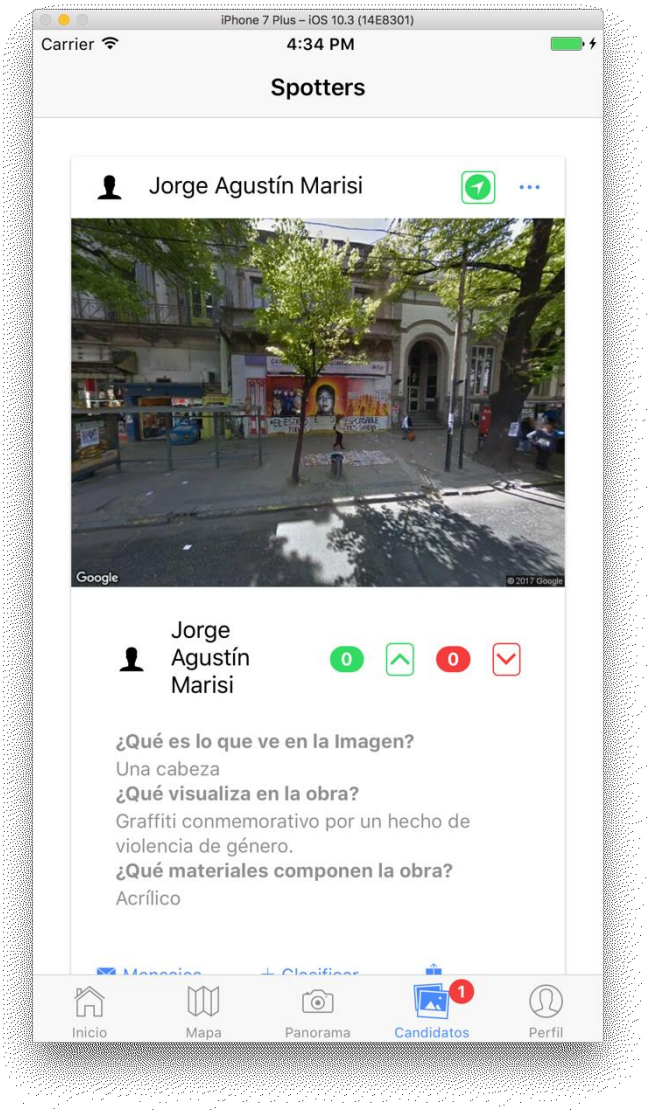

Ilustración 8 - Candidatos

La Ilustración 8 muestra la vista de Candidatos, donde pueden verse los candidatos que se encuentran activos en el área donde se encuentra centrado Google Street View.

Por cada candidato en la lista, se muestra el nombre del usuario que lo marcó, una fotografía del candidato, sus clasificaciones y los comentarios de los usuarios.

La fotografía del candidato puede reemplazarse por una vista embebida de Google Street View centrada en el candidato, para facilitar el análisis por parte de los usuarios y los expertos.

Las clasificaciones muestran las respuestas a cada pregunta del proceso de clasificación, el nombre del usuario que creó la clasificación y la cantidad de votos positivos y negativos que han sido otorgados por los usuarios.

Desde este listado, los usuarios pueden además crear nuevas clasificaciones para un candidato, votar sobre las clasificaciones ya creadas y sobre los comentarios de los otros usuarios y escribir nuevos comentarios.

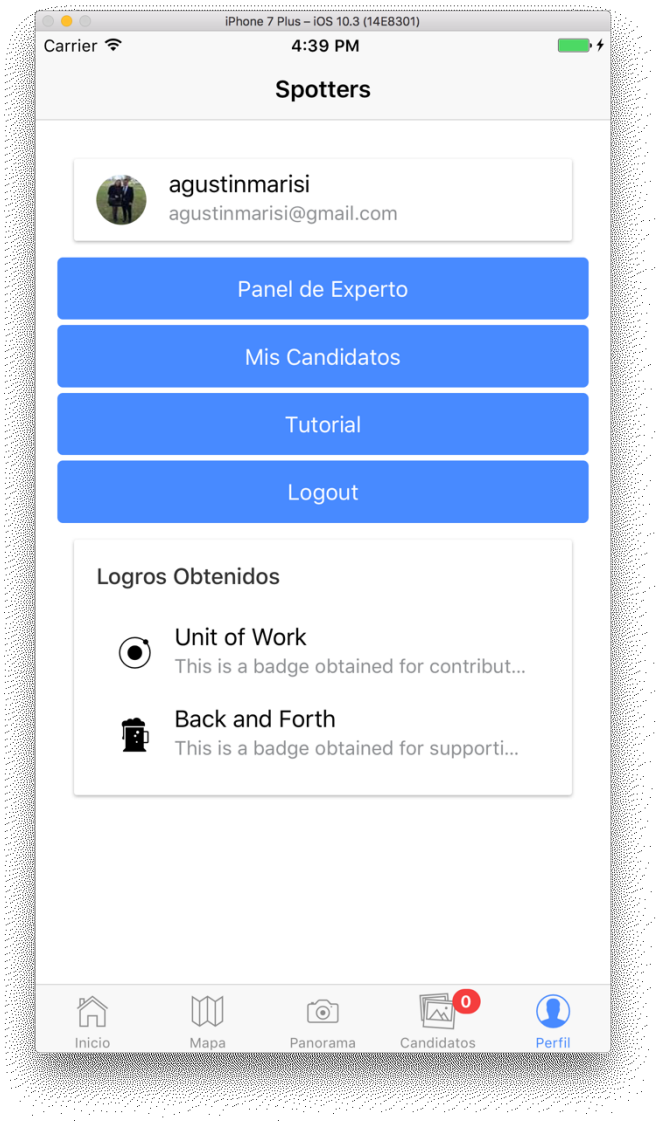

Ilustración 9 - Perfil del Experto

La Ilustración 9 muestra el perfil de un usuario con privilegios de experto. Puede observarse que existe una opción adicional, con el nombre *Panel de Experto*, el cual permite acceder a una vista con acciones específicas para observar los candidatos marcados por los usuarios y establecer si los mismos son válidos o no, además de realizar todas las acciones normales que pueden realizar los usuarios desde la vista de Candidatos.

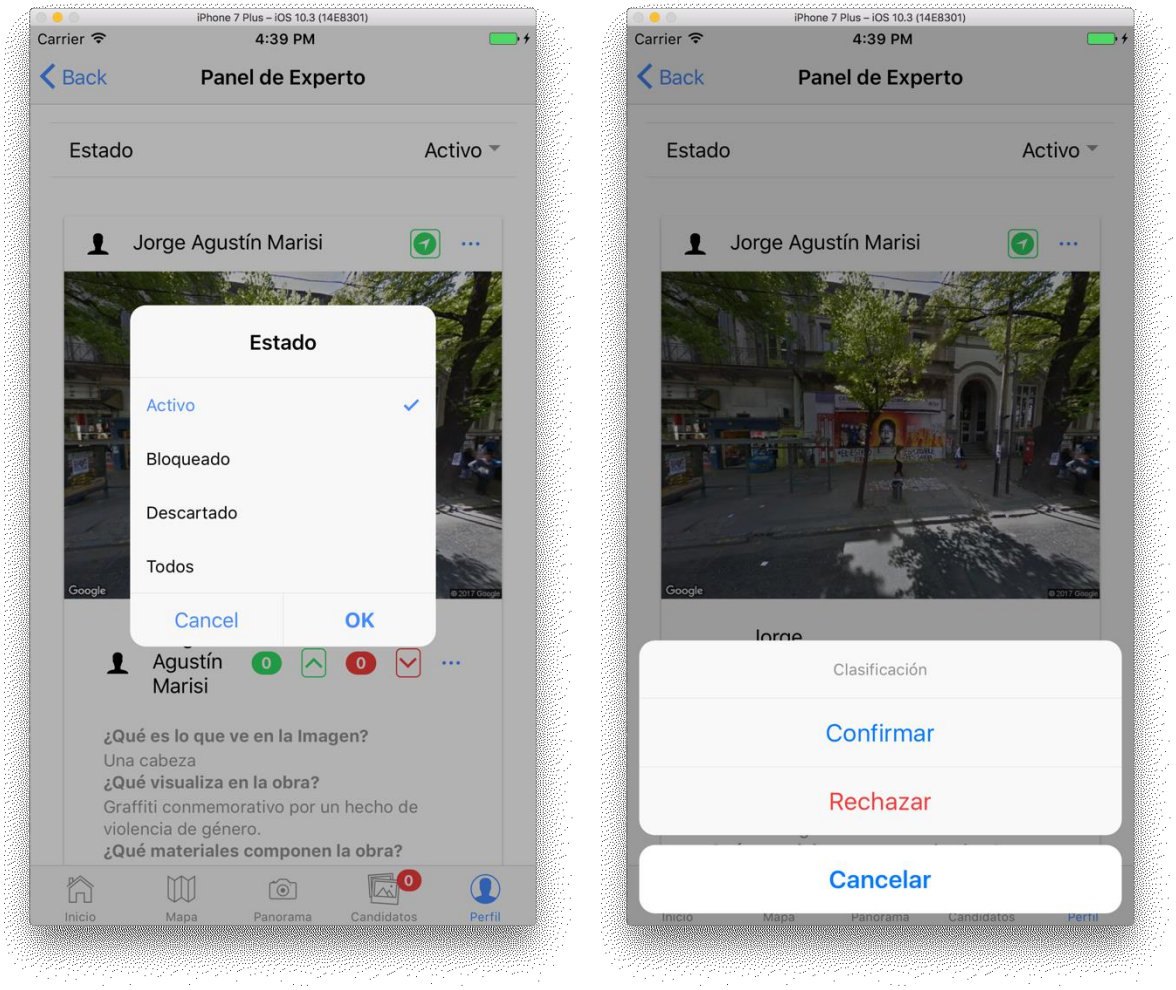

Ilustración 10 - Panel de Experto

La Ilustración 10 muestra las acciones disponibles desde el Panel del Experto, que permiten filtrar los candidatos de acuerdo a su estado y un menú contextual que permite confirmar o rechazar un Candidato.

6. Arquitectura del Framework

En este capítulo se desarrollan diversas decisiones que se tomaron a la hora de estructurar Spotters.

6.1. Configurable o Extensible

Al momento de comenzar a trasladar los conceptos de Spotters a un prototipo, se presentó como interrogante cuál sería la forma que tendría el programador de utilizar el framework para poder incluir las cuestiones particulares de su dominio. Analizando diversas alternativas, se encontraron dos caminos posibles: *Configuración* y *Extensión*.

La alternativa de Configuración consiste en proveer un mecanismo que permita derivar el comportamiento de la instancia de Spotters a partir de información contenida en un archivo con directivas o una base de datos. La alternativa de Extensión, consiste en proveer un conjunto de clases abstractas que el programador deba sub clasificar para crear las clases concretas necesarias para el funcionamiento de su instancia, tales como las preguntas que se realizan para la clasificación y los pasos del tutorial.

Como ventaja de la estrategia de Configuración se puede destacar la simplicidad de utilización, que permite que el desarrollador use el framework sin tener el conocimiento técnico de qué sucede en el núcleo, ya que no necesita modificar el código fuente de Spotters para tener una instancia en funcionamiento.

La estrategia de Extensión trae una dificultad extra para los programadores, que es conocer en detalle todas las clases que involucran al framework, que métodos implementan y cuales son aquellos que deben re implementarse para obtener el comportamiento deseado. Si bien es mandatorio que exista documentación del framework para que este sea usable, la curva de aprendizaje es más elevada y se necesita de un desarrollador con conocimientos específicos en el lenguaje de programación y las herramientas utilizadas para poder implementar una instancia funcional en un tiempo razonable. Por estos motivos, se decidió a utilizar el método de Configuración en lugar del de Extensión.

Respecto de las alternativas para la implementación de la instancia de Spotters configurable, se evaluó utilizar un mecanismo mediante archivos de marcado, como

XML, de serialización de datos, como YAML o JSON, o mediante base de datos, ofreciendo un backend para la creación de las instancias de clase necesarias.

La solución con archivos de configuración (XML, YAML o JSON) requiere que se creen esquemas para la validación de los datos. Posteriormente a la validación, debe analizarse el contenido de los archivos y finalmente realizar la carga de los mismos en una base de datos o realizar la lectura en tiempo de ejecución, sin utilizar persistencia. Otra posible dificultad que presenta este esquema es que debe existir documentación exhaustiva que le permita al desarrollador conocer las opciones de configuración y qué impacto tienen las mismas en la instancia, con lo cual se requiere un cierto nivel de estudio antes de poder tener una instalación funcional.

Nuevamente se le dio prioridad a la solución considerada como la más simple para el programador que fuera a utilizar Spotters y menos propensa a fallas. Por este motivo, nos decantamos por brindar un backend de carga de datos de referencia. Esto no sólo facilita la tarea del programador, sino que la haría más dinámica y ágil que tener que escribir un archivo estructurado, el cual debía pasar las validaciones que se mencionaron anteriormente.

Con esto en mente, también sorteamos el problema de qué pasaría si el programador modifica los archivos de configuración luego de haber realizado la configuración inicial: ¿Estos deberían volver a validarse y cargarse? ¿Se reemplazarían los datos existentes? Y toda una serie de interrogantes que harían que el framework fuera inconsistente, porque la respuesta correcta tal vez no fuera sólo una y estaría atado a criterios e interpretaciones personales.

6.2. Single Instance o Multi Instance

Desde la concepción de Spotters, se tuvo en cuenta la posibilidad de que fuera de instancia múltiple, es decir, que en una única instalación del framework pudiera convivir más de un proyecto de investigación, cada uno de ellos aislado del resto a nivel de la base de datos.

Este camino fue explorado durante la implementación de un primer prototipo, pero la tarea resultó no ser trivial respecto de la complejidad requerida para que cada proyecto quede aislado del resto y además se veían reducidas las posibilidades de

personalización, tanto a nivel del comportamiento del núcleo del framework, para el caso de que se requiriera personalizar el código para algún proyecto, como a nivel de las vistas del usuario, debido a que todos los proyectos debían tener un estilo uniforme y los cambios en uno terminarían por afectar al resto.

6.3. Modelo del Framework

A continuación, puede apreciarse el diagrama de clases de Spotters:

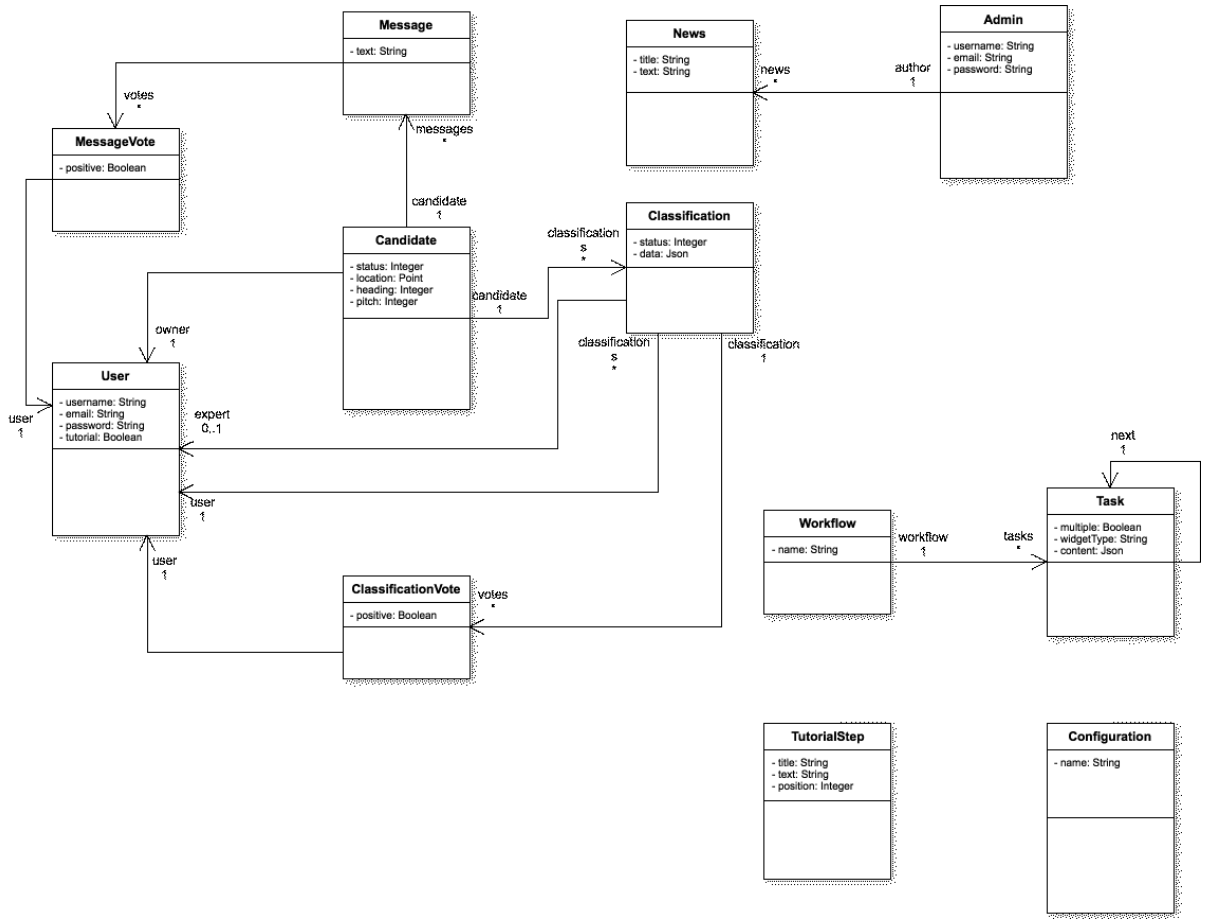

Ilustración 11 - Diagrama de Clases de Spotters

Las entidades más significativas que componen Spotters son:

- *Candidate*: es una de las entidades centrales del framework y representa un elemento marcado por un usuario. Tiene asociada información de geolocalización, datos del usuario que lo identificó, datos del experto que dio su veredicto sobre el punto, las clasificaciones creadas por los usuarios para el punto y los mensajes aportados por los usuarios.

- *Workflow*: representa un conjunto de preguntas que se utilizan para catalogar a un Candidate.
- *Task*: es una pregunta en particular, perteneciente a un Workflow, y utilizada como parte de la clasificación de un Candidate. Cada Task contiene datos de representación, la pregunta y las posibles respuestas, si las hubiera. *WidgetType* puede ser *text* o *choice*, para representar preguntas con respuesta abierta o cerrada. El flag *multiple* indica si cuando se elige un WidgetType choice, el usuario puede seleccionar más de una respuesta. Un Task tiene también el identificador de la pregunta que le sigue en el cuestionario. Si WidgetType es choice y multiple es falso, es posible indicar una pregunta siguiente para cada posible respuesta. De este modo es posible representar una estructura de árbol o grafo para encadenar las preguntas.
- *Classification*: representa la respuesta de un usuario a las Tasks de un Workflow. Contiene también un estado que representa si fue aceptado o rechazado por un experto.
- *ClassificationVote*: es un voto de un usuario sobre una Classification, para indicar si está de acuerdo o no con las respuestas dadas por el usuario que creó dicha Classification.
- *Message*: es un mensaje aportado por un usuario para un Candidate.
- *MessageVote*: es un voto de un usuario sobre un Message, que indica si está de acuerdo o no con dicho Message.
- *User*: es la representación de un usuario, ya sea jugador o experto, en el sistema. Posee Candidates, Classifications, Messages y votos.
- *Admin*: es la representación de un usuario desarrollador o administrador de una instancia de Spotters. Es el perfil del encargado de la configuración de la instancia.
- *News*: representa las novedades que se muestran en la portada del frontend sobre el proyecto.
- *TutorialStep*: representa un paso del tutorial. Se muestran según el orden indicado por el atributo *position*.
- *Configuration*: contiene variables de configuración de la instalación de Spotters, tales como punto de inicio, título o nombre de la instancia, entre otras.

6.4. Interacción de los Componentes

Spotters se encuentra dividido en dos componentes principales. Por un lado, una aplicación cliente, que puede tomar la forma de una aplicación web tradicional o de una aplicación nativa para plataformas móviles. Por el otro lado, un servidor que provee información para ser consumida por el o los clientes.

Ambas aplicaciones se comunican entre sí por medio de peticiones HTTP tradicionales (GET, POST y PUT), gracias a una API REST provista por el backend. La aplicación cliente no almacena datos en ninguna base de datos local, sino que toda la información se recupera desde el backend (noticias, pasos del tutorial, preguntas, candidatos existentes, etc.) y se almacena en el backend (nuevos candidatos, comentarios, etc.).

En el caso particular de algunas peticiones HTTP que crean o modifican datos en el servidor, se debe asegurar que el usuario que realiza la operación se encuentre autenticado. Para estos casos, no sólo se realiza un control de la operación desde la interfaz gráfica, sino que se utiliza *JWT*[18] para asegurar que el usuario se encuentra autenticado y que es quien dice ser. JWT o JSON Web Tokens es un método que permite la transmisión segura de información. Cuando el usuario inicia sesión en la plataforma, se le asocia un *token* que resultará válido por un lapso pre establecido de tiempo, con la posibilidad de ser renovado. Cuando el usuario realiza al servidor una petición que requiere autenticación, se adjunta el token en el encabezado de dicha petición y se valida en el servidor para verificar que sea correcto y que no haya caducado. De este modo, la sesión del usuario sólo se mantiene en la aplicación cliente y la aplicación servidor no necesita recibir datos de usuario y contraseña, sino que confía en el token recibido del lado del cliente.

Por otro lado, la aplicación cliente interactúa sin participación del servidor con otra serie de componentes externos, tales como la herramienta utilizada para la provisión del mapa y el panorama, y la utilidad para la autenticación de usuarios, analizadas ambas en el capítulo siguiente.

[18] https://jwt.io/

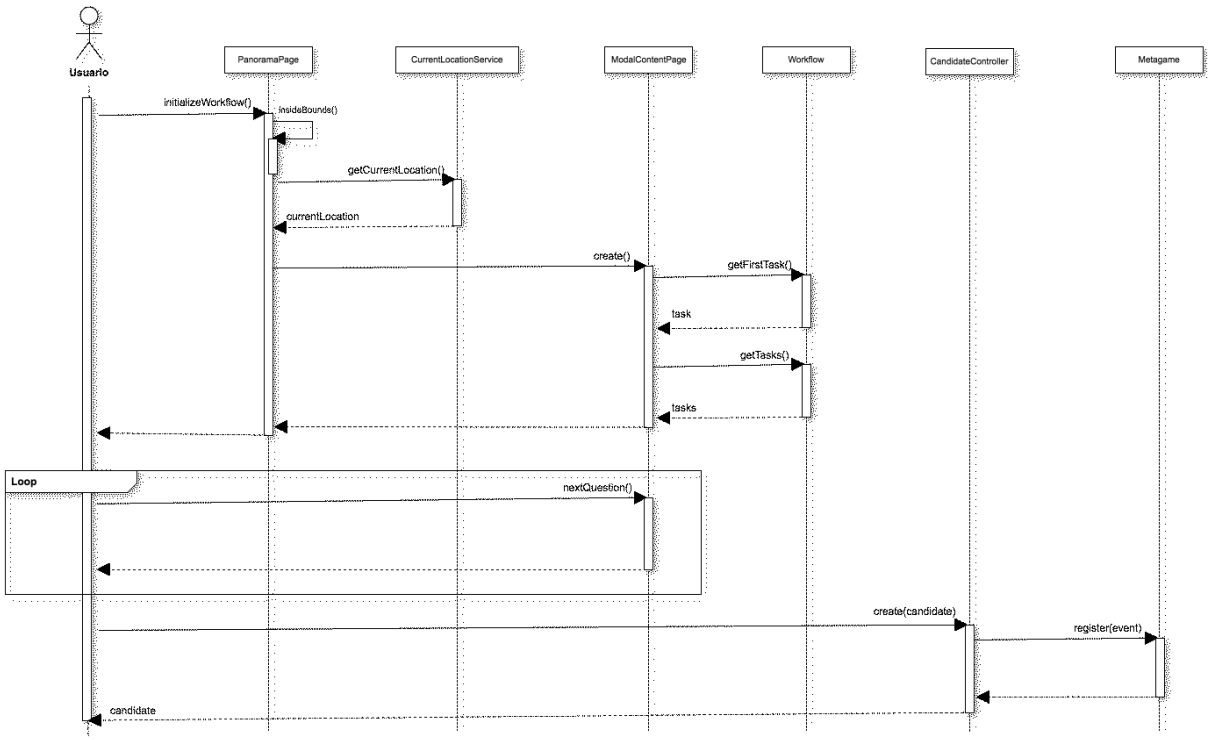

Ilustración 12 - Diagrama de Interacción

En la Ilustración 12 puede verse como los componentes interactúan entre ellos para crear un nuevo candidato y una clasificación para el mismo. El usuario interactúa con una serie de componentes del frontend, tales como *PanoramaPage* y *ModalContentPage*, para poder definir una clasificación sobre la imagen que está visualizando. Estos componentes se encargan de realizar una interacción interna entre ellos, obteniendo la información necesaria para que el usuario conteste una serie de preguntas en un orden específico (Workflow y Task). Una vez que el usuario ha respondido todo el cuestionario, esta información es enviada a un controlador específico del backend (*CandidateController*) para que el mismo realice las validaciones pertinentes y finalmente persista la información en la base de datos. Al mismo tiempo, este controlador realiza la comunicación con el Metajuego, mediante la clase *Metagame*, informando que un nuevo evento se ha producido en Spotters.

6.5. Interoperabilidad con Cientópolis

Tal como se mencionó en capítulos previos, la funcionalidad de Gamification de Spotters fue centralizada en una aplicación específica de Cientópolis, denominada Metajuego. Esta aplicación se encarga de recolectar información importante de todas las aplicaciones del ecosistema de Cientópolis, y presenta un juego didáctico en el cual se

puede ir avanzando de nivel y obteniendo distintos logros o recompensas por la utilización de los sistemas que lo componen. Estos logros se otorgan gracias a un análisis inteligente sobre los datos recolectados.

De esta manera, la interoperabilidad entre Spotters y Cientópolis está dada a través de una clase específica llamada Metagame. Esta clase se invoca cada vez que un evento que es de interés para el Metajuego ocurre en Spotters, informando qué tipo de evento ha ocurrido y qué usuario fue quien lo originó.

Desde el perfil del usuario de Spotters pueden visualizarse todos los logros obtenidos por el usuario, mostrando información detallada de cada uno de los mismos. Esta información se recupera a mediante una API provista por el meta juego, brindando de esta manera otra forma de interoperabilidad con Cientópolis.

6.6. Base Única de Usuarios

A partir del desarrollo de Spotters, se comienza a usar como herramienta de gestión de usuarios *Auth0*[19], una herramienta que provee SSO (*Single Sign On*) y autenticación por token como un servicio.

Se eligió este servicio porque desde Cientópolis se quiere ofrecer un ecosistema compuesto por múltiples aplicaciones donde sea posible ingresar a las mismas utilizando un mecanismo de inicio de sesión homogéneo, utilizando un único usuario global. De esta manera, el usuario no debería registrarse múltiples veces, ni recordar distintos usuarios, sino que habiendo iniciado sesión alguna vez en alguna de las aplicaciones, ya podría acceder a todas las demás sin tener que volver a registrarse o iniciar sesión.

Mediante Auth0 es posible configurar diversas aplicaciones que utilicen una base común de usuarios entre sí, y que permitan inicio de sesión utilizando diversos mecanismos ofrecidos por aplicaciones de terceros, como Twitter, Facebook, Github, entre muchos otros. También permite utilizar mecanismos más tradicionales, como usuario y contraseña en una base de datos propia.

[19] https://auth0.com

Spotters es la primera de las aplicaciones de Cientópolis que implementa el acceso mediante Auth0, pero está en los planes que otras aplicaciones lo incorporen, debido a su simplicidad de configuración, administración y variedad de guías de configuración para múltiples lenguajes de programación y frameworks, tanto server-side como client-side.

Desde el lado de Spotters, se obtiene de Auth0 la información que el usuario ingresó en la red social que se utiliza para validar su identidad (Facebook, por ejemplo). Así, es posible mostrar su avatar, nombre de usuario, correo electrónico, entre otros datos. También se accede al identificador de usuario aportado por la red social, el cual se utiliza para crear una entidad usuario local en Spotters, la que se asocia al resto de los objetos que la necesitan, como candidatos, clasificaciones, mensajes, etc.

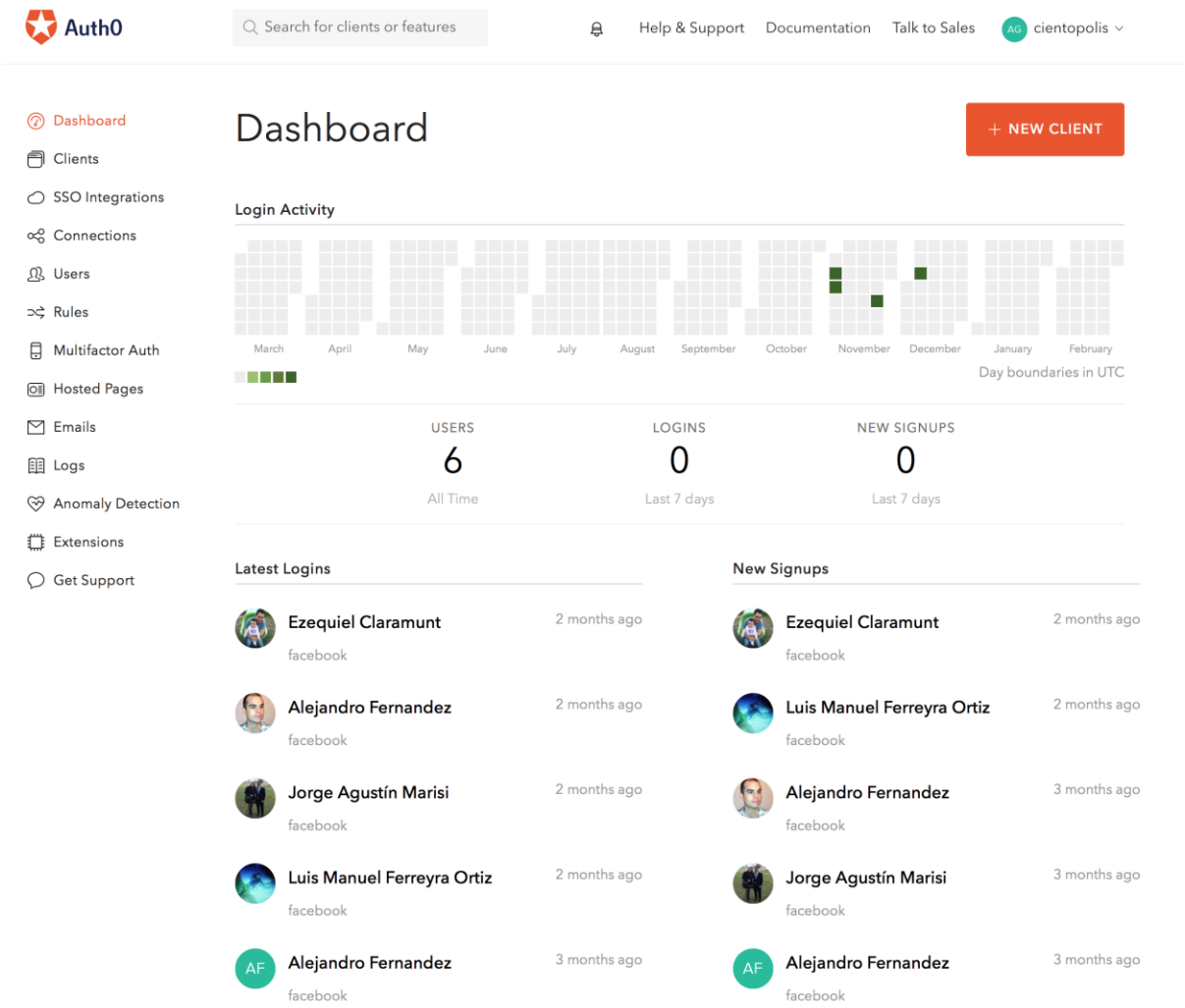

Ilustración 13 - Dashboard de Auth0

En la Ilustración 13 se observa el dashboard general de Auth0, donde es posible visualizar estadísticas de acceso, configurar aplicaciones clientes que utilicen la base de usuarios, configurar los mecanismos de inicio de sesión de entre una lista de varias decenas, activar la autenticación de dos factores, gestionar extensiones creadas por terceros, entre otras opciones.

6.7. Instanciación y HotSpots

Tal como fue mencionado en secciones previas, la estrategia para crear una nueva instancia de Spotters consiste en configurar un conjunto específico de instancias de clases del framework.

Para la creación de estas instancias, se desarrolló un backend específico. De ese modo, el programador no necesita tener conocimientos específicos de los lenguajes de programación utilizados en la creación de Spotters, sino que sólo debe dar de alta una serie de entidades en dicho backend, utilizando formularios que validan que la configuración creada sea correcta.

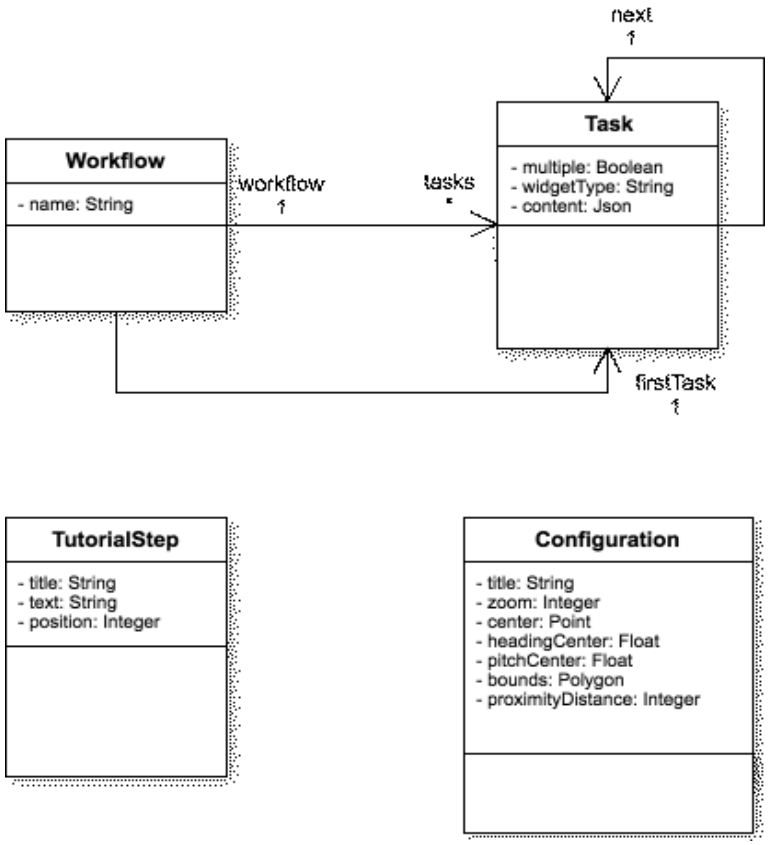

Ilustración 14 - HotSpots de Spotters

En la Ilustración 14 se visualizan las clases para las cuales es necesario crear instancias específicas al momento de realizar la instanciación del framework.

En primer lugar, es necesario crear una única instancia de la clase Configuration desde el backend. Tal como se mencionó previamente, esa clase define el comportamiento básico de una instancia de Spotters, como qué nombre se va a mostrar en el frontend y cuál es el área de influencia de la aplicación en el mapa.

Otra de las clases de las cuales es necesario crear instancias para el correcto funcionamiento del framework es TutorialStep. Se deben definir tantas instancias como pasos se desee que posea el tutorial, indicando para cada una una posición, que se utilizará para mostrarlas luego ordenadas en el frontend.

Finalmente, el último de los HotSpots está compuesto por las clases Workflow y Task. Estas dos clases se relacionan entre sí y conforman el mecanismo de creación de clasificaciones para un nuevo candidato. Las particularidades de cómo configurar una Task ya han sido desarrolladas previamente, pero en líneas generales debe definirse el contenido de la pregunta, las posibles respuestas y cuál puede ser la o las posibles preguntas siguientes.

7. Plataforma de Instanciación

7.1. Instanciación y Configuración del Framework

Para la instanciación del Framework debe partirse de una copia del código fuente del backend. Por simplicidad, se ofrece junto con el código fuente un proyecto de *Vagrant*[20] que permite mediante *VirtualBox*[21] y *Ansible*[22] tener un entorno básico configurado y funcional en unos minutos, aunque también podría realizarse una configuración manual y así evitar la necesidad de utilizar una máquina virtual.

A grandes rasgos, lo que se realiza mediante Vagrant y Ansible es la instalación de todos los paquetes necesarios, creación de los archivos de configuración y puesta en funcionamiento del servidor web del backend.

Una vez que se encuentra corriendo el servidor web, es necesario ingresar al panel de administración, que fue creado para simplificar la creación de las instancias de clases necesarias.

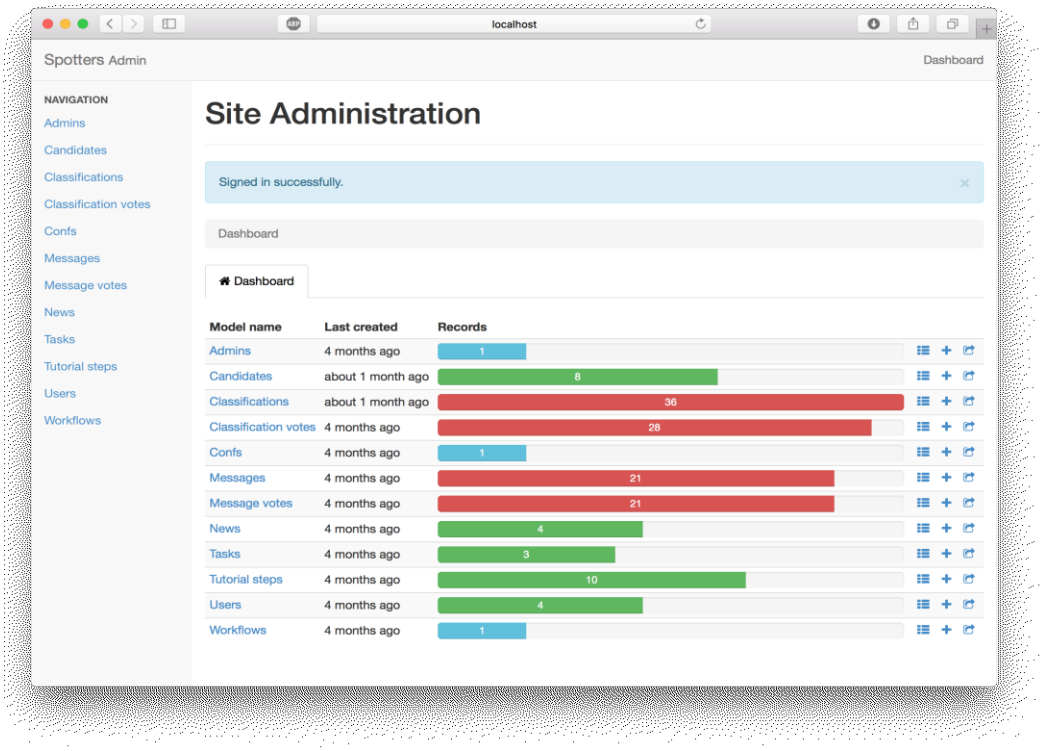

Ilustración 15 - Portada del Backend

[20] https://www.vagrantup.com/
[21] https://www.virtualbox.org/
[22] https://www.ansible.com/

Desde el backend, cuya interfaz puede verse en la Ilustración 15, es necesario crear instancias de las entidades para el correcto funcionamiento de Spotters.

En primer lugar, es necesario realizar la configuración general de la instancia, desde *Confs*. Allí deben introducirse datos como el título que se verá en el frontend, punto inicial en el mapa si no está disponible el GPS, niveles de zoom por defecto y cuadrante de influencia de la aplicación. Al utilizar el proceso de generación automática antes descripto, ya se genera una instancia de la configuración con datos de ejemplo, que deberán modificarse para ajustarlos a la nueva instancia. Por el contrario, si se está generando una instancia nueva, debe crearse una nueva instancia de la clase Confs. En la Ilustración 16 puede observarse una captura del formulario de configuración.

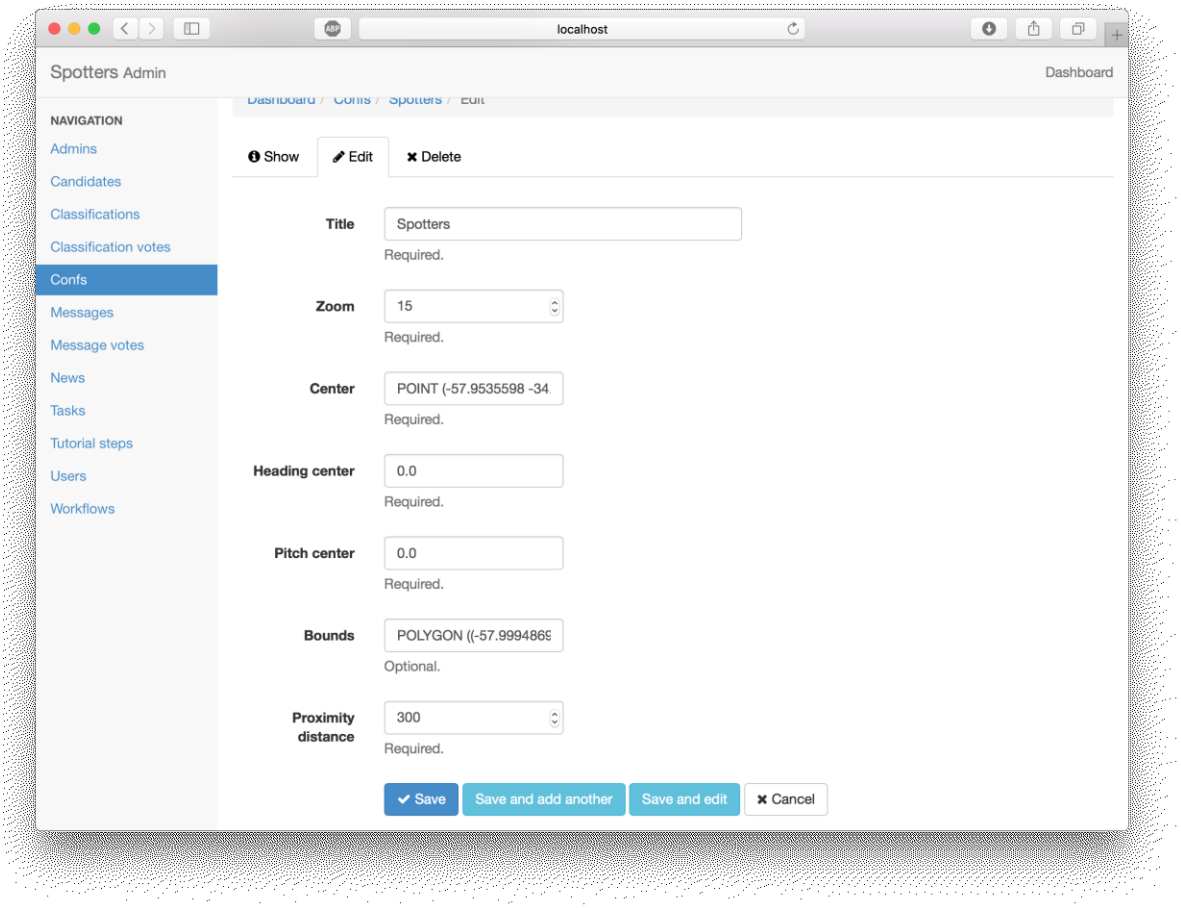

Ilustración 16 - Configuración de la Instancia

A continuación, deben crearse los pasos del tutorial de entrenamiento para nuevos usuarios, desde *Tutorial steps*. Debe generarse una instancia por cada paso del tutorial, indicando para cada una en qué orden se encuentran. Al momento de

visualizarlos desde el frontend, se ordenarán de forma ascendente de acuerdo al campo position.

Finalmente, es necesario crear instancias de las clases que componen el proceso de clasificación de los puntos de interés. En primer lugar, debe crearse un Workflow, cuya utilidad es la de englobar al conjunto de preguntas e indicar cuál es la primera pregunta del proceso. Luego se deben crear tantas instancias de Task como sean necesarias para el proceso de clasificación, indicando las características específicas de cada pregunta, como su enunciado, tipo de respuesta y opciones. Ejemplos del formato requerido para los campos de formulario de Tasks pueden encontrarse en el código fuente, ya que algunos campos utilizan texto estructurado en formato JSON.

7.2. Roles

En esta sección se describen los distintos roles de usuario presentes en Spotters.

7.2.1. El Rol del Usuario

El usuario de Spotters es quien tiene como tarea la localización y clasificación de puntos de interés. Puede moverse por el mapa a la vez que tiene acceso a una vista panorámica de 360º del punto en el cual se encuentra centrado el mapa.

Desde la vista panorámica, un usuario puede marcar un punto de interés. Cuando se dispone a marcar un punto de interés, el usuario deberá responder a un cuestionario, el cual incluye preguntas que son de interés para los expertos y que dan como resultado una clasificación.

Esta clasificación está sujeta a la votación de otros usuarios, que estén de acuerdo o no. Es posible que otro usuario cree otra clasificación para un punto de interés ya marcado, si considera que ninguna de las existentes es adecuada.

Además de poder marcar un punto de interés y clasificarlo, los usuarios pueden debatir sobre dichos puntos. El debate puede darse en forma de mensajes o votos sobre los mensajes que consideren importantes.

7.2.2. El Rol del Experto

El experto es quien se vale de los puntos de interés marcados por los usuarios para su investigación, y es también el responsable de aceptarlos, rechazarlos y elegir la clasificación más adecuada, si la hubiera.

Puede realizar las mismas actividades que un usuario normal, pero sus comentarios en los hilos de discusión estarán destacados, por ser la palabra autorizada en la materia de estudio de la instancia particular de Spotters.

Tiene a su cargo la tarea de visualizar los puntos marcados por los usuarios e indicar si las clasificaciones que indicaron son correctas.

También tiene acceso a estadísticas sobre el proyecto y a toda la información en bruto que se ha recolectado, para su posterior uso en las tareas de investigación que emprenda.

El experto también tiene la tarea de realizar un tutorial introductorio para los usuarios finales, donde se le explique cómo encontrar los puntos de interés, cómo detectar falsos positivos o qué tener en cuenta al momento de clasificar un elemento de interés. Este tutorial es presentado después del registro de un nuevo usuario.

Finalmente, también es quien debe generar los cuestionarios que deben responder los usuarios al clasificar un punto de interés, ya que es quien tiene en claro qué información espera obtener del proyecto y qué datos le interesan.

7.2.3. El Rol del Administrador/Programador

El administrador es una instancia de Spotters es quien tiene a su cargo la instalación, configuración y ampliación de una instancia, la gestión de los datos, administración de cuentas de usuario y mantenimiento en general de la instancia.

La instalación y configuración de una instancia de Spotters ya fue explicada en la sección anterior. Una vez que la instancia está funcional, es responsabilidad del administrador mantenerla en ese estado, verificando que no haya problemas con los usuarios, realizando las actualizaciones sobre el código base si hubiera nuevas versiones de Spotters, y controlando el estado del servidor en el que se ejecuta.

8. Implementación

En este capítulo se abordan los detalles técnicos de la implementación de Spotters, tales como tecnologías utilizadas para la aplicación cliente, herramientas utilizadas en la confección de la API que conforma el lado del servidor, herramientas de base de datos y demás APIs de las que depende y con las que interactúa el framework.

8.1. Server-side

El backend o capa del servidor de Spotters es una aplicación desarrollada mediante el framework *Ruby on Rails*[23], y que conforma una API REST con la cual interactúa la aplicación que es vista por el usuario final. Esto hace posible que un único backend reciba peticiones de diversas aplicaciones cliente, tales como una versión web tradicional o aplicaciones móviles nativas.

Rails provee una herramienta de línea de comandos que permite generar de forma rápida el scaffolding (modelo, controladores y vistas por defecto) de cada entidad del sistema que se desee. Si bien posteriormente es necesario un trabajo de ajuste y configuración sobre cada scaffolding generado, esto permite tener una aplicación con un conjunto de funcionalidad básica desde el primer instante.

Gracias a la gema *JBuilder*[24], se simplifica la serialización a JSON de las entidades de la aplicación que deben ser servidas ante cada petición a la API. También se utiliza la gema *Devise*[25] para proveer un mecanismo de inicio de sesión seguro a los administradores.

Para la configuración de las instancias de Spotters por parte de los desarrolladores, se utiliza la gema *rails_admin*[26], la cual provee un gran número de funcionalidades que son de utilidad al sistema, por ejemplo, la posibilidad de crear scaffolding de formularios de los modelos, brindando validaciones client-side de los campos de cada una de las entidades, así como los campos relacionados a dichos modelos. Otra de las funcionalidades que es de utilidad es la posibilidad de filtrar los elementos en el listado por cualquier tipo de campo y cualquier criterio. Además,

[23] http://rubyonrails.org/
[24] https://github.com/rails/jbuilder
[25] https://github.com/plataformatec/devise
[26] https://github.com/sferik/rails_admin

provee exportación de las entidades en distintos formatos, tales como JSON, CSV o texto plano.

Con Rails como backend se utiliza un motor de base de datos con un esquema relacional, concretamente *PostgreSQL*[27].

Para simplificar el proceso de instalación, configuración y puesta en producción del backend, y poder sanear las diferencias de configuración en los entornos de desarrollo de cada programador, se utiliza Vagrant junto con Ansible. Mediante estas dos herramientas es posible generar mediante virtualización un entorno de trabajo uniforme y repetible, ya que el servidor de aplicación queda corriendo en una máquina virtual y la instalación de cada uno de sus componentes se realiza mediante una serie de pasos pre configurados, los cuales pueden repetirse una cantidad arbitraria de veces con la seguridad de que se obtendrá siempre el mismo resultado.

8.2. Client-side

Dado que es necesario que la vista con la cual interactúan los usuarios y expertos sea dinámica, debido a la interacción con el mapa/panorama y las características de marcado y votación, se utiliza *Angular*[28] en conjunto con *Ionic*[29].

Tal como fue mencionado en la sección previa, el frontend de Spotters es una aplicación creada mediante Ionic, que utiliza Angular y *TypeScript*[30]. Ionic permite crear aplicaciones que pueden visualizarse desde la web o de forma nativa en Android, iOS y Windows Phone. Las aplicaciones creadas con Ionic pueden considerarse como *PWAs* o *Progressive Web Apps*, ya que proveen características que son consideradas como normales en aplicaciones nativas, pero no siempre están presentes en aplicaciones web, tales como notificaciones, acceso offline, procesos en segundo plano, entre otras características.

El cliente de Spotters es una aplicación creada con la filosofía *mobile-first*, en la cual se intenta que, en primera instancia, la aplicación sea funcional en dispositivos móviles, dejando las computadoras tradicionales en un segundo plano. Se eligió esta

[27] www.postgresql.org
[28] https://angular.io/
[29] https://ionicframework.com/
[30] https://www.typescriptlang.org/

metodología debido a la amplia predominancia de Smartphones y Tablets que puede apreciarse actualmente como dispositivos de acceso a Internet.

Ionic provee un catálogo de componentes específicamente adaptados para dispositivos móviles, pero que también pueden verse de forma adecuada mediante un navegador web tradicional de escritorio. Dicho catálogo de componentes incluye fuentes especiales, íconos, botones, listas de elementos, wizards (asistentes), entre otros, los cuales brindan, a pesar de ser componentes HTML, la apariencia de componentes nativos de cada plataforma, sin la necesidad de desarrollar de forma específica una aplicación nativa para cada una.

Angular permite interconectar las distintas vistas y controladores que conforman la aplicación Ionic, de modo tal que no es necesario recargar la página para pasar de un componente a otro. También permite interactuar con la API del backend de forma asincrónica, con lo cual se evita que la interfaz se congele mientras espera que lleguen los datos del servidor.

Angular, como su predecesor AngujarJS, es un framework para la creación de aplicaciones web dinámicas. Permite aplicar los conceptos del patrón de diseño MVC [Gamma1993] al frontend de una aplicación, donde el modelo está representado por objetos que enmascaran peticiones a una API REST (el backend), las vistas son templates HTML y los controladores son clases encargadas de interconectar el modelo con los templates y que almacenan lógica. Este framework también provee inyección de dependencias [Fowler2004] y data-binding entre los templates y el modelo para la actualización automática de las vistas ante cualquier cambio. Como característica adicional, Angular es mobile-first y al igual que Ionic, provee facilidades para la creación de PWAs.

TypeScript es un súper set de JavaScript creado por Microsoft. Este dialecto de JavaScript intenta corregir ciertas falencias del lenguaje que derivan en malos hábitos, a la vez que agrega chequeo estático de tipos. El código TypeScript se procesa mediante un compilador, el cual genera código JavaScript compatible con los navegadores. Dicho compilador además presenta los errores de tipos que pudieran existir, a la vez que facilita y agiliza la escritura de programas por parte de los desarrolladores, ya que es

posible corregir una abundante cantidad de errores al momento de la etapa de escritura y se evita así, en ciertos casos, tener que hacer debugging durante la ejecución.

Para acelerar el desarrollo con TypeScript y evitar la necesidad de realizar el proceso de compilación manualmente, existen editores e IDEs que incorporan soporte mediante plugins, y que permiten visualizar los errores a la vez que se escribe. Ejemplos de editores con soporte para TypeScript en la actualidad son: VisualStudio Code, Atom e IntelliJ Idea.

8.3. Base de datos

Respecto del motor de base de datos utilizado por Spotters, ya se ha mencionado que el backend utiliza PostgreSQL.

La utilización de este motor de base de datos en particupar por sobre otras alternativas relacionales, como MySQL, responde únicamente a la mayor integración que provee Ruby on Rails con PostgreSQL mediante ActiveRecord, la cantidad de gemas que se encuentran en desarrollo activo para interactuar con dicho motor de base de datos, y la mayor madurez de las extensiones para trabajo con información geoespacial que existen para PostgreSQL.

Para la representación de información geoespacial en la base de datos se utiliza *PostGIS*[31], que extiende la funcionalidad existente en el motor de base de datos para soportar nuevos tipos de datos que permiten representar puntos geográficos, rutas o cuadrantes, y funciones que permitan operar sobre dichos datos, como cálculo de cercanía de puntos e intersecciones de polígonos.

Específicamente, los tipos aportados por PostGIS se emplean para representar la ubicación de los Candidates en el mapa, poder seleccionar dichos Candidates según la cercanía al punto en el cual se encontraba centrado el panorama, y también para acotar el marcado de puntos a un cuadrante en particular, como por ejemplo, la ciudad de La Plata.

[31] http://postgis.net/

8.4. Google Maps API

Spotters utiliza la API de Google Maps para mostrarle al usuario el mapa de la región que se desea que explore.

Mediante la utilización de la API, es posible crear y mostrar un mapa centrado en un determinado punto, con un determinado nivel de zoom y una determinada vista, que puede ser del mapa tradicional, una vista de satélite, o una versión híbrida.

Mediante la API también se agregan marcadores o *markers* al mapa, que representan cada uno de los Candidates que se encuentran en una ubicación cercana a la figura de Street View. Estos marcadores pueden personalizarse, para que, en función de distintos parámetros, el ícono que se muestre sea diferente o posea etiquetas descriptivas. De esta manera, es posible indicarle de manera visual a un usuario de Spotters que un Candidate que él ya había visitado y votado ha sufrido cambios (una nueva clasificación) o que hay nuevos Candidates que él todavía no ha visto.

8.5. Google Street View API

La API de Street View se utiliza para presentarle al usuario una interfaz desde la cual pueda recorrer el área de interés como si estuviera caminando realmente por esa región.

Desde esa API se capturan coordenadas, ángulo e inclinación de lo que está visualizando el usuario cuando desea marcar un nuevo Candidate, y esos son los datos que se registran en el backend.

También se utiliza para determinar en qué punto se encuentra el usuario en el mapa y poder así recuperar los Candidates cercanos, de acuerdo a la configuración de proximidad que se haya establecido.

Mediante la API de Street View, también es posible evitar tener que realizar capturas de pantalla de lo que visualiza el usuario, ya que provee una URL desde la cual, para una serie de parámetros, como latitud, longitud, ángulo e inclinación, se obtiene la imagen pertinente que se observaría desde Street View al acceder a dicho punto.

9. Caso de estudio: Bellas Artes

En este capítulo se detallan los requerimientos y experiencias obtenidas en el desarrollo de una aplicación real utilizando el framework Spotters: Identificación de obras de Arte en la vía pública para la Facultad de Bellas Artes.

9.1. Identificación de Obras de Arte

Como se mencionó en capítulos anteriores, uno de los proyectos que fomentan el desarrollo de Spotters y brindó el marco teórico/práctico para la realización de esta tesina de grado surge de la Facultad de Bellas Artes.

Desde esta Facultad se presenta un proyecto con el objetivo concreto de identificar, clasificar y catalogar las distintas obras de arte que están dispersas por todo el territorio de la ciudad de La Plata [FerreyraOrtiz2016], ya que no existe ningún tipo de documentación concreta que las identifique y detalle donde se encuentran posicionadas, ni su estado de conservación, desde el año 2001 a la actualidad. Una vez encontradas, las obras deben ser clasificadas según los materiales de construcción y el tipo de obra de arte que es, por ejemplo, si es una escultura o un monumento, si es de mármol, de cemento o de ladrillos, y desde el punto de vista de qué muestra o representa.

De esta manera, gracias a las tecnologías de Google Maps y Google Street View es posible permitir a los usuarios que utilizan la aplicación la posibilidad de moverse por las distintas calles que componen la ciudad, y brindarles una forma fácil de marcar una obra de arte que estén visualizando.

Una vez que el elemento es encontrado, se despliega un conjunto de preguntas para completar, que tienen como objetivo la clasificación de la obra de arte que se está observando. De esta manera, es posible desplegar distintas preguntas en función de las respuestas que brinda el usuario a una pregunta previa, lo que permite obtener clasificaciones más concretas. Así, se obtiene una primera aproximación a la Ciencia Ciudadana ya que múltiples usuarios que pueden o no desarrollarse en un ámbito científico, colaboran entre sí para realizar tareas que pueden ser consideradas repetitivas con el objetivo de brindar datos para un proyecto de investigación.

Una vez que ya se cuenta con elementos clasificados, se puede encontrar un segundo punto de contacto con la Ciencia Ciudadana, ya que es necesario realizar un primer filtro para que los expertos en la materia no analicen puntos de interés que no son obras de arte, o aquellas cuya clasificación es incorrecta, ya sea por equivocación a la hora de responder a las preguntas, o por el accionar de usuarios malintencionados. Por lo tanto, todos los usuarios pueden realizar votaciones sobre las clasificaciones existentes en la plataforma o realizar una nueva clasificación en caso de que crean que la existente no es adecuada.

En caso de que existan múltiples clasificaciones o que un experto en la materia decida que vale la pena involucrarse en una clasificación en particular, Spotters le brinda la posibilidad de realizar comentarios sobre lo que se está discutiendo en una obra de arte. Los mensajes del experto se muestran destacados, para que el resto de los usuarios tengan en consideración que se trata de alguien relacionado con el proyecto.

Como cualquier tarea que involucra Ciencia Ciudadana, el objetivo fundamental es obtener datos sobre la realización de distintas tareas concretas para su posterior uso. Por esto, desde el panel de administración al cual se accede desde el backend, se provee la posibilidad de visualizar las distintas obras marcadas en el mapa y descargar los puntos donde fueron encontradas con su respectiva clasificación para futuros análisis.

9.2. Deployment de la Aplicación

Debido a que la aplicación fue realizada íntegramente con Spotters, el deployment se realizó utilizando las herramientas de configuración que provee el mismo framework para poder instanciar cada una de las aplicaciones necesarias.

Como se mencionó anteriormente, la instanciación se realiza en dos etapas; primeramente, se descarga el código fuente y se configuran elementos fundamentales para el correcto funcionamiento de la aplicación, tal como base de datos a utilizar y contraseña de la misma.

Una vez que la aplicación se encuentra corriendo, es posible realizar la segunda etapa para terminar con la configuración del nuevo sistema, y es a través de un panel de administración que provee Spotters en el backend. Este panel le permite configurar

múltiples cuestiones relacionadas con el caso en concreto, como, por ejemplo, nombre de la aplicación, workflow de preguntas, área de Google Maps a utilizar, etc.

Concretamente, se crearon las siguientes instancias particulares:

- Una instancia de un Workflow para contener las preguntas.
- Tres Tasks (preguntas) con sus correspondientes respuestas, encadenadas entre sí. Se comenzó por la última, ya que es la que no debe relacionarse con ninguna otra.
- Se modificó el Workflow creado en primer lugar para asociarlo a la primera Task del cuestionario.
- Se crearon las instancias de los TutorialSteps, donde se explicaba a los usuarios cómo identificar y clasificar puntos de interés.
- Se creó una instancia de Confs con los datos específicos de la aplicación, siendo los más relevantes: nombre a mostrar, punto donde se centrará el mapa por defecto, nivel de zoom por defecto y área de influencia de la aplicación (un vector donde cada posición contiene la latitud y longitud de un vértice del polígono que compone el área de influencia).

Una vez que esta configuración fue finalizada desde el backend, se realizó la configuración del frontend, donde fue necesario indicar la URL del backend para que obtenga la configuración.

Desde la URL https://www.cientopolis.org/spotters/ es posible acceder a un video ilustrativo del proceso de instalación, además de enlaces a los repositorios donde se encuentran las distintas partes del código fuente que componen a Spotters.

9.3. Tiempo Requerido Para la Instanciación

Para el cálculo de los tiempos requeridos para que quede configurada una instancia de Spotters, se dejan de lado los tiempos utilizados en la descarga de los componentes, ya que son dependientes de la velocidad de conexión a internet, así como también se omiten los tiempos requeridos para configurar el servidor para que se encuentre disponible desde internet. Sólo se consideran los tiempos puros utilizados para la configuración de una nueva instancia utilizando el framework, tanto para el frontend como para el backend.

Una vez que se cuenta con el código fuente del backend disponible en el equipo que será el servidor, se requiere aproximadamente una hora para la creación de la base de datos que utilizará para el almacenamiento de la información de la aplicación, carga de los datos iniciales, configuración de la conexión con el Metagame y deploy del backend para que se ejecute y sea accesible por vía web.

Con el backend accesible, se deben cargar los datos de Spotters que luego serán accesibles desde el frontend: Workflow, Tasks, TutorialSteps y Confs, las cuales fueron descriptas previamente. Si bien el tiempo requerido para esta etapa puede variar de acuerdo a la cantidad de preguntas y pasos que pudiera tener el tutorial, se considera que puede demorar aproximadamente dos horas para la finalización de la carga de los datos.

Finalmente, se debe realizar la configuración y deploy del frontend, para que se conecte al backend que fue configurado anteriormente. Esta etapa puede llegar a demorar aproximadamente una hora.

En conclusión, la instanciación de una nueva aplicación con Spotters puede tener una duración aproximada de cuatro horas, teniendo como resultado final una aplicación completamente funcional.

10. Conclusiones

Durante el desarrollo de este trabajo se propuso crear un framework que simplifique la creación de aplicaciones de Ciencia Ciudadana, contribuyendo de ese modo a enriquecer el ecosistema de herramientas libres y de código abierto disponibles.

Originalmente, se consideró que las aplicaciones resultantes del uso de dicha herramienta tuvieran elementos de Gamification, aunque durante el proceso se decidió delegar ese comportamiento a otro proyecto que se encuentra en desarrollo actualmente dentro del ámbito de Cientópolis.

Como resultado concreto, se logró construir un framework que, mediante una serie de configuraciones básicas, genera aplicaciones de Ciencia Ciudadana utilizando Google Maps y Google Street View, sin la necesidad de que quien utiliza esta herramienta posea conocimientos de programación.

Durante el desarrollo del framework, se incorporaron tecnologías que simplifican aún más la tarea de quien lo utiliza a la hora de hacer el deployment de una aplicación, particularmente Vagrant y Ansible, que permiten tener un backend funcional mediante la ejecución de un conjunto muy reducido de comandos.

Finalmente, se implementó de manera exitosa una de las aplicaciones concretas planteadas al principio de este trabajo, la de la Facultad de Bellas Artes, la cual se desarrolló en un corto lapso de tiempo y sin realizar modificaciones sobre el código del framework.

11. Trabajos Futuros

En este capítulo se detallan las funcionalidades o características que a futuro sería posible agregar a Spotters, junto con la posible integración a otros sistemas pertenecientes al proyecto Cientópolis.

11.1. Detecciones de Usuarios Malintencionados

Debido a que las aplicaciones generadas con Spotters serían, en principio, públicas, y que las mismas se encontrarían accesibles a través de internet, es necesario proveer algún tipo de seguridad para que usuarios con malas intenciones no realicen un mal uso de la plataforma, registrando puntos de interés que no son válidos o escribiendo comentarios poco apropiados en las áreas de interacción social, lo que terminaría afectando la calidad de los datos de los que dispondría el investigador.

Los usuarios malintencionados son un problema habitual en cualquier tipo de software, ya sea público o cerrado, y el tipo de daño que pueden provocar varía en función de qué tan profundo puedan acceder, ya sea a la capa de aplicación o incluso al servidor. Por ejemplo, un usuario de estas características podría robar datos sensibles de los usuarios, acceder a la base de datos, modificar el código fuente o dejar offline el sistema modificando la configuración del servidor que lo almacena. Estos tipos de ataques están mitigados hoy en Spotters mediante diferentes técnicas de seguridad. Algunas de dichas técnicas son:

- La contraseña de la base de datos se encuentra encriptada en el servidor.
- La seguridad de las contraseñas de los usuarios (longitud, utilización de caracteres especiales, etc.) está asegurada mediante la imposición de ciertas reglas al momento del registro.
- La utilización de OAuth como mecanismo de registro e inicio de sesión evita que se almacenen contraseñas en nuestra base de datos y se delega la seguridad a Auth0.
- El servidor cuenta con filtrado de puertos por IP Tables en una configuración de lista blanca y se bloquea cualquier IP excepto que se encuentre entre la lista de IP de confianza. En dicha lista se encuentran sólo los desarrolladores y administradores del servidor.

- Se realizan chequeos de seguridad buscando intrusiones en el código fuente.
- Se bloquean ataques de SQL Injection gracias a las características del ORM utilizado por Ruby on Rails, ActiveRecord.
- Se bloquean ataques XSS (*Cross Site Scripting*) en los campos de entrada de usuario (los datos proporcionados por los usuarios son filtrados de código malintencionado).
- Si se detectan patrones de acceso potencialmente peligrosos, como el caso de un ataque DoS o DDoS, se bloquean automáticamente las direcciones IP que están perpetrando el ataque para que no puedan acceder al servidor web.
- Se realizan backups periódicos y de forma automatizada de la base de datos.

Sin embargo, Spotters no cuenta por el momento con ningún mecanismo de seguridad que evite que un usuario registre puntos de interés que en realidad son falsos, que realice comentarios ofensivos o sin sentido, que realice clasificaciones sin analizar realmente las características de lo que está visualizando, o que utilice las características de votación para dejar un feedback negativo en casos que no lo ameritan.

Es por esto que sería necesario sumar mecanismos de control del comportamiento de los usuarios, que permitan detectar dichas situaciones y suspendan o sancionen al usuario que está haciendo un mal uso del sistema, antes de que provoque un daño más severo.

También sería deseable que dicho mecanismo, al detectar un usuario malintencionado, anule las acciones realizadas por este.

Para la detección de este tipo de usuarios, podrían considerarse como variables la cantidad de votos emitidos, la cantidad de puntos de interés marcados, la cantidad de clasificaciones contestadas, todo ello en un corto lapso de tiempo.

11.2. Estadísticas de Uso

Actualmente, el dashboard del experto de Spotters brinda únicamente información de los puntos que se marcaron, sus clasificaciones y comentarios. No se realiza ningún procesamiento adicional sobre esa información, que le permita al experto o al administrador de la instancia obtener datos más específicos sobre los usuarios y su comportamiento.

Sería importante realizar una mejora en este sentido para lograr un mayor nivel de comprensión de los usuarios que utilizan la plataforma y de esta manera poder decidir qué nuevas características priorizar para brindar una mejor experiencia de uso.

A partir de los datos que registra Spotters, podría deducirse qué tiempo permanecen activos los usuarios en promedio en cada sesión, cuántos puntos de interés registran, quiénes prefieren las características sociales por sobre las de exploración o cuáles son las zonas calientes del mapa (*heatmap*), entre otras.

Teniendo estos datos, podría promoverse el recorrido de ciertas zonas del mapa de las cuales se cuenta con menos información o promover más la comunicación con los expertos, si los usuarios se ven más motivados por el debate.

11.3. Integración con Samplers

Una de las problemáticas que tiene actualmente la implantación de Spotters es que se basa puramente en Google Maps y Google Street View; es decir, que los panoramas que se muestran no volverán a renovarse hasta que Google envíe nuevamente vehículos a recorrer la región.

Para los casos de ejemplo presentados en esta tesina, el factor de actualización no resulta extremadamente relevante ya que, por ejemplo, los monumentos que existen en la ciudad no reciben una renovación periódica. Sin embargo, para el ejemplo de las rampas o vados peatonales, la actualización podría ser importante ya que se pueden omitir rampas o vados que fueron construidos con posterioridad a la toma de las imágenes.

Por este motivo, podría considerarse integrar Spotters con Samplers, el proyecto de Cientópolis mencionado en el Capítulo 3, permitiendo que los usuarios suban sus propias imágenes con coordenadas cuando les sea posible, sin depender de los panoramas de Street View. De esta manera, si un usuario desea adjuntar una fotografía más actualizada que la que se ve en la aplicación, podría utilizar su teléfono celular para realizar dicha tarea.

11.4. Integración con Panoptes Talk

Panoptes Talk[32] es una herramienta que provee la comunidad de Panoptes, y se encarga de gestionar los comentarios de los usuarios sobre una fotografía, video, o cualquier otro elemento dispuesto para el análisis de la comunidad.

Esta herramienta es de código abierto y permite crear hilos de discusión a través de una API.

El mecanismo de hilos de discusión que provee Spotters en la actualidad fue diseñado únicamente para ser utilizado dentro de la aplicación, y no provee características avanzadas que sí incorpora Panoptes Talk, como, por ejemplo, la utilización de Hashtags, y acceso programático a los hilos y mensajes por medio de una API.

La ventaja adicional que proveería la utilización de una API es que podría centralizarse toda la gestión de comentarios de todos los proyectos de Cientópolis dentro de un único espacio y utilizando una única herramienta.

Al momento de desarrollar Spotters, se intentó realizar esa integración, pero no fue posible debido a la falta de documentación, el nivel de acople con Panoptes y lo acotado del tiempo de desarrollo de este trabajo.

11.5. Interacción en Tiempo Real y Notificaciones Push

En la versión actual de Spotters, si un usuario se encuentra posicionado sobre un punto de interés, y otro usuario realiza una acción sobre el mismo, no es posible visualizar dicha acción hasta que no se recargue la vista del primer usuario.

Para poder brindar esta funcionalidad se propone la implementación a futuro de WebSockets [Fette2011], que permiten refrescar el contenido de la vista de aplicación sin la necesidad de hacer *polling* sobre el servidor y sobre todo brindan la posibilidad de que el usuario visualice nuevo contenido sin tener que realizar ninguna acción particular.

[32] https://github.com/zooniverse/Talk-Api

Junto con Web Sockets, podría ser útil incorporar en los dispositivos que lo soporten *Notificaciones Push* [Bell2011], que le lleguen al usuario cuando un evento importante sucede dentro de Spotters, como, por ejemplo, un comentario en alguno de sus puntos de interés o una respuesta en la clasificación por parte de un experto.

Estas notificaciones podrían utilizarse también para volver a captar la atención de un usuario que hace un tiempo que no ingresa a la plataforma, en el caso de que Spotters se encuentre instalado como una aplicación nativa en un dispositivo móvil.

Bibliografía

- Horita, F. E. A., Degrossi, L. C., de Assis, L. F. G., Zipf, A., & de Albuquerque, J. P. (2013). The use of volunteered geographic information (VGI) and crowdsourcing in disaster management: a systematic literature review.
- Hosseini, M., Shahri, A., Phalp, K., Taylor, J., & Ali, R. (2015). Crowdsourcing: A taxonomy and systematic mapping study. Computer Science Review, 17, 43-69.
- Werbach, K. (2014). (Re)defining gamification: A process approach. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 8462, pp. 266-272). Springer Verlag.
- Hamari, J., & Eranti, V. (2011). Framework for designing and evaluating game achievements. Proc. DiGRA 2011: Think Design Play, 115(115), 122-134.
- Bhattacharjee, Y. (2005). Citizen scientists supplement work of Cornell researchers: a half-century of interaction with bird watchers has evolved into a robust and growing collaboration between volunteers and a leading ornithology lab. Science, 308(5727), 1402-1404.
- Bowyer, A., Lintott, C., Hines, G., Allan, C., & Paget, E. (2015). Panoptes, a Project Building Tool for Citizen Science. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15). AAAI, San Diego, CA, USA (pp. 1-2).
- Bowser, A., Hansen, D., He, Y., Boston, C., Reid, M., Gunnell, L., & Preece, J. (2013, October). Using gamification to inspire new citizen science volunteers. In Proceedings of the first international conference on gameful design, research, and applications (pp. 18-25). ACM.
- Deterding, S., Khaled, R., Nacke, L., & Dixon, D. (2011). Gamification: toward a definition. *Chi 2011*, 12–15. https://doi.org/978-1-4503-0268-5/11/0.
- Denny, P. (2013). The effect of virtual achievements on student engagement. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 763. https://doi.org/10.1145/2470654.2470763.
- Celasco, M., Yañez, J. I. (2015). Gamification en la búsqueda de galaxias.
- Herzig, P. (2014). Gamification as a service.

- Brabham, D. C. (2008). Crowdsourcing as a model for problem solving: An introduction and cases. Convergence, 14(1), 75-90.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1993, July). Design patterns: Abstraction and reuse of object-oriented design. In European Conference on Object-Oriented Programming (pp. 406-431). Springer Berlin Heidelberg.
- Fowler, M. (2004). Inversion of control containers and the dependency injection pattern.
- Ferreyra Ortiz, L. M. (2016). Inventario, catalogación y estudio diagnóstico del patrimonio escultórico urbano de la ciudad de La Plata.
- Fette, I. (2011). The websocket protocol.
- Bell, K. M., Bleau, D. N., & Davey, J. T. (2011). U.S. Patent No. 8,064,896. Washington, DC: U.S. Patent and Trademark Office.